NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

SIMULATION-BASED ANALYSIS AND EVALUATION OF TACTICAL MULTI-HOP RADIO NETWORKS

by

Howard D. Smith

March 2009

Thesis Advisor:	John Gibson
Thesis Co-Advisor:	Gurminder Singh

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** March 2009	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE Simulation-Based Analysis of Tactical Multi-Hop Radio Networks			**5. FUNDING NUMBERS**
6. AUTHOR(S) Howard D. Smith			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			**12b. DISTRIBUTION CODE**
13. ABSTRACT (maximum 200 words) For many years, the technologies involved in the newest generations of tactical communication equipment have increased the reliability and security of tactical voice communications from the highest to the lowest levels of combat command. However, the complexities inherent to wireless data networks have prevented the reach of valuable data links from extending efficiently and reliably to the lowest levels of tactical command. This thesis attempts to quantify the performance of tactical data networks using existing technologies and currently deployed mobile wireless networking devices by analyzing the results of network simulations involving currently deployed devices. By quantifying these performance metrics and comparing them to previously collected simulation results involving experimental technologies, we hope to provide a mode of comparison that will accurately reflect the degree to which newer mobile wireless networking devices will benefit our operational forces.			
14. SUBJECT TERMS Radio Networks, Multi-hop Networks, Computer Networks, Tactical Radio Networks, Software-Based Network Simulation, Network Analysis, SINCGARS, EPLRS, Cooperative Diversity, JCSS			**15. NUMBER OF PAGES** 125
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited.

SIMULATION-BASED ANALYSIS AND EVALUATION OF TACTICAL MULTI-HOP RADIO NETWORKS

Howard D. Smith
Captain, United States Marine Corps
B.S., United States Naval Academy, 2002

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2009

Author: Howard D. Smith

Approved by: John H. Gibson
Thesis Advisor

Gurminder Singh
Co-Advisor

Dr. Peter J. Denning
Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

For many years, the technologies involved in the newest generations of tactical communication equipment have increased the reliability and security of tactical voice communications from the highest to the lowest levels of combat command. However, the complexities inherent to wireless data networks have prevented the reach of valuable data links from extending efficiently and reliably to the lowest levels of tactical command. This thesis attempts to quantify the performance of tactical data networks using existing technologies and currently deployed mobile wireless networking devices by analyzing the results of network simulations involving currently deployed devices. By quantifying these performance metrics and comparing them to previously collected simulation results involving experimental technologies, we hope to provide a mode of comparison that will accurately reflect the degree to which newer mobile wireless networking devices will benefit our operational forces.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to thank my wife, Marcie, for the understanding and patience she showed while I was working on this project. She is a wonderful woman, and I am truly blessed to have her in my life. I would also like to thank my thesis advisor, John Gibson, for being so actively engaged in providing solid support and well thought out advice throughout each phase of this process. His help and guidance have heightened my academic horizons and made the creation of this thesis an enjoyable and positive experience. Lastly, I need to thank Byron Harder for the advice he provided while I was attempting to recreate portions of the simulations he used for his thesis research. His assistance was tremendously appreciated.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. OBJECTIVE

Our objective for this thesis is to model a variety of realistic network simulations, which demonstrate the performance of currently deployed wireless mobile networking devices, in a manner that is consistent with current tactical data network loads. We wish to demonstrate each network's ability to support the demands of a variety of different data traffic densities, and then compare the performance of each model to the results of a model used in previous thesis research simulations, involving an experimental communications device that implements some emerging wireless networking technologies. Most of our simulations will be roughly based on the simulations performed in [1], and after our analytical baseline is established, we will then discuss any performance gains or losses found within each type of network and explore the reasons for any divergences.

The main contribution of this research will be the quantification of specific data networking performance metrics for legacy mobile wireless networking devices, and the comparison of these results to the already quantified performance results from [1]. Our intent is to provide a more accurate understanding of the actual degree to which newer wireless mobile networking technologies can benefit our operational forces.

B. WHY DATA AT LOWER TACTICAL COMMAND LEVELS

The ability to provide data services to the lowest levels of tactical combat commands has existed for decades. However, the low available bandwidth and difficulty of implementation of the data services within currently deployed devices has severely limited the operational incorporation of data services outside the stationary command center. The problem with many of the radios being used by our military is that they were designed with voice communications as the primary focus, with data capabilities added as an inefficient secondary capability.

The development of improved wireless data networking technologies has enabled the creation of certain wireless networking devices that could effectively extend significant information flow outside of the stationary command center and allow unit commanders to share greater quantity and better quality of information with their lower (typically highly mobile) levels of command.

For example, with more effective data network links, it would be possible to quickly provide significantly more detailed targeting information (i.e., a photograph or live streaming video) to the smaller mobile units, which would reduce the probability of them targeting inappropriate buildings, vehicles or people.

Although the benefits of increased tactical data capabilities have been recognized for a long time, it has not been until recently that the technologies that promise to fill this communication gap have begun to mature to the point of possible implementation. With a relatively recent increase in the amount of commercial demand for mobile wireless voice and data services, we have seen an unprecedented growth in the development and refinement of more capable and reliable mobile networking technologies. Incorporating these new technologies into our tactical communication architecture will be crucial to our military's ability maintain our technological advantage, and it will greatly increase our tactical commanders' ability to more accurately ascertain real-time battlefield conditions, in order to more quickly make informed decisions during their execution of combat operations.

C. RESEARCH QUESTIONS

1. What kind of data traffic is required by units functioning at the lower tactical levels of command?

2. What are the capabilities and limitations of currently fielded wireless networking devices?

3. How does the ability of currently available mobile networking devices to handle various types of data network traffic loads compare to that of devices using newer types of wireless networking technologies?

4. What, if any, improved capabilities should be achieved by future devices, and are these improvements significant enough to justify the fielding of new equipment?

D. ORGANIZATION

In Chapter II, we will provide an overview of the seven-layer OSI Service Model, as presented in [14], with an emphasis on the characteristics of the data link layer, mobile multi-hop networks and some of the issues that make them so challenging to implement, and on one of the newly implemented technologies that holds some promise of overcoming some of the limitations of wireless networks at the link layer, which were previously modeled and analyzed in [1].

In Chapter III, we will explain the general functionality of the different radios to be modeled and the limitations of each radio's ability to provide the types of data network capabilities really needed at the tactical levels.

In Chapter IV, we will explain the characteristics and limitations of the Joint Communications Simulation System network simulation software we will use to analyze and evaluate the effects of various types of network traffic across existing tactical radios.

In Chapter V, we will compare the results of our simulations with those obtained from simulations involving a specific prototype device that was modeled under similar scenarios in previous thesis work.

In Chapter VI, we will summarize the overall comparisons between the performances of each of the simulated radios and make recommendations for future research related to our results.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND INFORMATION

A. OSI SERVICE MODEL ARCHITECTURE OVERVIEW

In order to provide some kind of structure to the development of different network protocols, current networking developers usually operate within the boundaries of commonly accepted service model architectures. One of the most common network service models is the seven-layered Open Systems Interconnection (OSI) service based architecture paradigm.

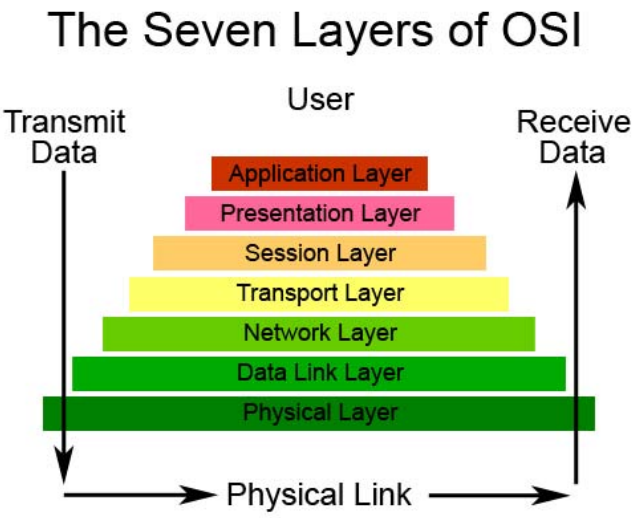

Figure 1. 7-Layer OSI Model (From University of Washington website).

As shown in Figure 1, this architecture divides all networking protocols into layers, or groups of services, based on the specific functions they help perform during the data networking process. Processes at each layer of service communicate with each other and work in conjunction to provide specific services to the layer above it. Each layer can only use the services provided by the layer below it, and this uniform flow of utility allows the functions performed within each layer to operate independently of the functions performed within any of the other layers. This modularity of function allows network designers greater flexibility in how they choose to combine different

implementations of services performed at each layer, without having to worry about a single change at one layer causing each of the other layers to stop functioning.

The first two layers, the physical and data link layers, and the functions they perform are of particular interest to the analysis performed in this thesis, since it is the establishment of reliable and efficient wireless data link connections that present the most significant challenges to the creation of data networks across mobile wireless nodes.

1. Physical Layer

The physical layer encompasses all protocols and functions performed by the physical medium by which a signal carrying network information is accomplished. This includes, but is not limited to radio waves traveling through the air, analog waveforms traveling across copper wire, or discrete light waves traveling across fiber optical wires.

For the purposes of this thesis, all physical layer transmission simulations will be radio waves being transmitted through the air between like communication nodes.

2. Link Layer

The link layer encompasses all protocols and functions performed between two communication nodes immediately before sending and immediately after receiving frames across the physical layer. The link layer's main purpose is to forward a network layer datagram through whatever types of transmission links exist along the path from the transmission's sender to its destination. If there are different types of links along this path, then the link layer will perform different tasks accordingly, in order to accommodate the needs of each type of link. The different tasks performed at different types of links are transparent to the processes running at the network layer.

Some common tasks performed by processes running at the link layer are encapsulation, controlling overall link access, ensuring reliable delivery, controlling the flow of frames across each link, and error detection/correction of each frame. All of these processes work towards the common purpose of ensuring that each transmitted frame is reliably received by each communicating node without errors. It also attempts to

ensure that each transmission does not interfere with the transmissions of other nodes communicating across the same physical medium, in a manner that permanently denies another sender access to network resources.

It is the functions performed at the link layer that create the most unique types of complexity, especially when there are many communication nodes attempting to share access to the same wireless transmission medium. Wireless links are much more susceptible to signal errors caused by the surrounding environment (electromagnetic radiations, competing communication transmissions, multipath interferences, etc.), so effective error detection and correction protocols are much more important, and can be significantly more difficult in wireless networks than in wired networks.

There are also generally less wireless bandwidth resources available to meet the throughput needs of host applications, so an application (or a suite of multiple applications) that may run very well on a wired network, may perform poorly when run across a wireless medium. While wired networks can always add more wires to increase the number of channels available for signal transmission, wireless networks are limited to the use of a fixed amount of available radio frequency spectrum. It is because of these difficulties that the functions performed at the link layer are the most relevant to the different communications nodes evaluated later in this thesis, and are discussed in more detail below.

a. Encapsulation

Encapsulation is an important concept, because different types of networked devices may perform this task very differently from one another. Encapsulation is used by the link layer to create a data frame that is formatted appropriately for transmission directly across the physical layer data transmission medium.

Services belonging to the link layer will receive a datagram from the network layer and add an additional header and trailer to it, in order to create a properly formatted link layer frame. This means that the actual frame being transmitted across the physical medium is different from the datagram that will ultimately be received by the

different layer protocols running on the receiving devices. This is because the information required to forward each frame across each communication link is not a concern of the higher layered protocols, so it is not generated or shared outside of the link layer. Also, since the size of each encapsulated frame can have a significant effect on throughput and retransmission requirements for each frame, some link layer services may divide the network layer frames into smaller sized fragments, which will each be encapsulated and transmitted across the physical layer separately, only to be pieced back together at the receiving node. The smaller each fragment is, the more likely it will be able to reach its destination without errors from signal interference.

However, it is not always the case that smaller fragments are desirable for a given data link. If the same amount of overhead is required to forward each fragment, then with an increased quantity of fragments being forwarded, there will be more total overhead introduced across each link. This will decrease the overall throughput of a data link. So, smaller fragment sizes may not be suitable for hosts that require links with higher throughputs. These kinds of considerations create complexities for network managers, who must balance the throughput needs with the acceptable level of errors/retransmissions based on the needs of the supported hosts.

b. Error Detection/Correction

Error detection is an important function performed by the data link layer protocols because it is typically at the physical layer where most frame errors occur. This is especially true for networks operating across wireless communication links, since the quantity and types of interference inherent in wireless waveform propagation far exceed the types of interference affecting wired waveform propagation.

Error detection is typically performed using a checksum value or a Cyclic Redundancy Check (CRC) included in the trailer of the link layer frame, which is created by the transmitting node for every unique frame transmitted, and used by the receiving node to validate the integrity of each received frame. If an error is detected, the frame can be retransmitted at the link layer level, without requiring the higher level protocols to detect and retransmit the erroneous frame.

Alternatively, many link protocols use types of forward error correction schemes, such as fountain coding, which not only allow the host to detect an error in a corrupted fragment, but also allow potential correction of these fragments. This method could be desirable over more simple error detection schemes, since it eliminates the need for these corrupted fragments to be retransmitted across the link. While this method of error correction requires some additional processing capabilities on behalf of the receiving host, the main drawback to the use of these schemes is in the additional overhead required for their implementation, and the decrease in link throughput it causes.

c. Media Access Control

Media Access Control (MAC) is a protocol that attempts to govern the manner in which each networked communication node accesses the transmission medium. Various types of MAC protocols have been developed that attempt to perform this task in different manners. The main goal of a MAC protocol is to ensure that each node attached to the network is allowed to transmit its signal across the transmission medium as quickly as possible, while producing minimum interference with the transmissions of other nodes attempting to access the same transmission medium.

Balancing the desire for high throughput at each node with the minimization of overall transmission collisions is especially challenging when working with wireless networks. Since two-way wireless links are half-duplex by nature (assuming all nodes are only communicating across a single channel), wireless networks cannot effectively use the more efficient MAC protocols that are so common in the full-duplex wired networks.

Additionally, unlike wired networks, there is no guarantee that every wireless node can detect the transmissions of all other nodes in the same wireless network. As shown in Figure 2, the “hidden node” problem, can render an intermediary node incapable of relaying either one of the two transmitting node’s transmissions. If both node A and node B are not within range of each other, then the Hub node may receive both signals at the same time and not be able to understand either one of them.

This means that it is possible for two nodes to attempt to access the same network resources at the same, without ever realizing that their transmissions are interfering with each other.

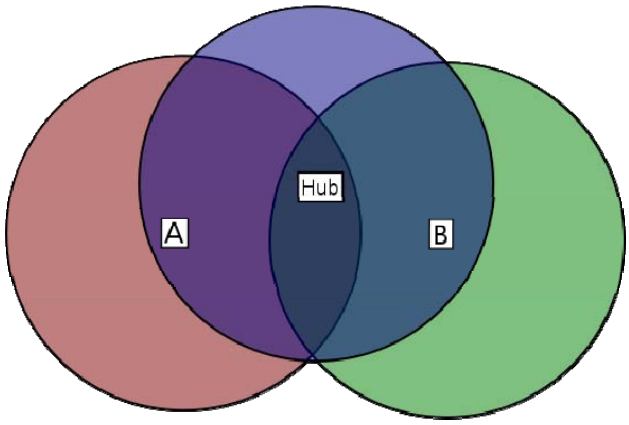

Figure 2. Hidden Node Problem (From Wikipedia.org).

B. MULTI-HOP MOBILE NETWORKS

A multi-hop mobile network is network composed of more than two mobile nodes, where it is possible for two of the nodes within the network, which are not in direct contact (i.e., they are out of transmission range of each other), to send and receive transmissions between each other through separate nodes that belong to the same network.

1. How Multi-hop Networks Work

In a multi-hop network, in order for a signal packet to travel from the originating node to the destination node it may be necessary for the packet to travel through more than one hop or transmission time slot. In order to accomplish this, the packet must be relayed through a separate node (or multiple separate nodes) that acts as a relay.

For example, in Figure 3 the center node, whose transmission radius is represented by the red (center-most) circle, is not within transmission range of the

leftmost node, whose transmission radius is represented by the blue circle. In order for a packet to travel from the center node to the leftmost node, it must be relayed through the node whose transmission range is represented by the yellow (upper-center) circle. The yellow node acts as a relay node that belongs to the same network as the other two nodes, but is neither the originator nor final recipient of the packet. Thus, since the source and destination nodes are separated by more than a single hop and can still communicate with each other, it is considered to be a multi-hop network.

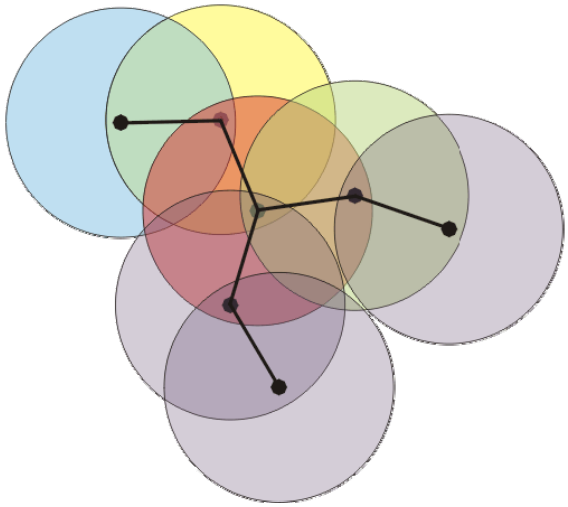

Figure 3. Multi-hop Network Node Diagram (From RWTH-AACHEN University website).

Multi-hop networks can be very useful when a particular node needs to communicate with a node that is outside of its own transmission range, and can greatly expand the overall communications range of any node within a single multi-hop network. As long as the originating node is within range of another node that is either linked directly or indirectly with the destination node, it can still send the packet with a reasonable expectation that it will eventually arrive at the desired destination.

2. Complexities Inherent To Multi-hop Networks

a. Infinite Loops

In a single-hop or infrastructure-based wireless radio network, if a node receives a packet that is destined to a different node it will simply ignore it. In a wireless multi-hop network, the receiving node may relay the packet in an attempt to deliver the message to some distant recipient that might not be within range of the transmission's originator. If there are more than one relaying nodes, there is the possibility that these intermediary nodes will continue to exchange these retransmissions indefinitely, producing a kind of infinite receive-retransmit cycle. This will not only reduce the available bandwidth to all nodes within range of these transmissions, but will unnecessarily tie up processing resources of each node involved in the loop. This is similar to issues experienced in early Ethernet networks, and was solved by assigning each packet a TTL number, which is decremented by each routing node that retransmits the packet. This same type of solution can be implemented in a multi-hop wireless by introducing a maximum hop limit for each transmission.

Another method of mitigating infinite loops is the implementation of a controlled network flooding scheme. In this scheme, each packet received by a node, but is not destined for that node, is relayed once and only once. This ensures that each packet will be transmitted across the entire area of network coverage, but avoids the potential for infinite loops, since a node encountering the same packet more than once will simply drop it. The main problem encountered in this type of scheme is in the implementation of a method that allows each node to accurately distinguish between copies of previously encountered packets and packets that it has not yet processed.

Controlled flooding can be implemented in two ways: it can use the packet header to store the accumulated routing data created as each packet is relayed through the network, or it can require each node to cache packet data locally, and compare the cached data to each received packet. In heavily trafficked networks, the additional overhead of adding routing data to each packet header may be undesirable due to the potential for increased network congestion. Additionally, asking each node to

maintain a cache of previously seen packets increases the storage and processing burden on each node device, which may place an undesirable amount of strain on the device's power and hardware resources.

b. Hop Limits

Placing a maximum limit on the number of hops a packet can travel before being discarded may eliminate the possibility of infinite transmission loops forming between two nodes, but it also introduces limitations to the overall coverage area of the network, so this limit must be chosen carefully. If the hop limit is too low, this could significantly reduce the transmission area coverage benefits gained by implementing a multi-hop network. If the hop limit is too high, there can still be significant drain on network and node resources, similar to that incurred from infinite transmission loops. Therefore, the maximum hop limit must be set to a level that effectively balances the needs to the network users and the limitations of the network nodes. At a minimum, the hop limit should be at least slightly greater than the expected diameter of the network. If it is smaller than the network diameter, then it is possible for a packet to be dropped before it even has a chance to travel far enough reach its destination. If the hop limit is significantly greater than the network diameter, then problems with creating network congestion may occur.

c. Reduced Bandwidth

All wireless nodes utilize the same physical transmission medium to transmit any type of network traffic. So, if each node spends time retransmitting traffic they did not originate, then it means they must spend less time transmitting their own traffic. The necessary overhead of retransmission traffic in multi-hop networks takes up bandwidth that cannot be used by other nodes to initiate their own transmissions, so this potentially effectively reduces the bandwidth available to any one node, assuming it is not the only node allowed to originate transmissions.

When a network node transmits a signal destined to another node that is more than one hop away, the transmitting node must wait at a minimum for the closest

relaying node to retransmit the signal, before transmitting its next segment. If this does not happen, the second transmission will interfere with the relaying of the first.

In some cases, if a network has a maximum hop count of *n*, each transmitting node will wait *n*-1 time slots between transmissions, in order to ensure that no collisions occur between its next transmission and the transmissions of nodes that are still relaying the original message. This type of transmission delay is used by the EPLRS device discussed in later chapters, and really only benefits mobile networks whose nodes may shift between being tightly bunched and spread far apart, as it would ensure that collisions are not cause by relay transmissions when all network devices are within transmission range of each other. While this may help decrease the chance of interference between transmissions, it also decreases each node's available throughput to 1/*n* of the total number of time slots.

d. Additional Processing Required

While the implementation of a wireless multi-hop network seems simple in theory, there are certainly greater processing and buffering requirements introduced by the need for each individual network node to perform a greater number of tasks related to processing each received transmission. Each node has the additional computational burden of needing to analyze each received transmission and recognize if that transmission is destined to itself or to another node, and perform some form of queuing process for the message to be retransmitted and, then retransmit it at the same time that it may be trying to transmit its own messages. While these tasks are no different than those of routers in a wired network, in a wireless network all hosts must perform these actions. These actions may have additional power requirements for which existing hardware is not well suited and could introduce mobility restrictions by requiring larger batteries (more weight) or less distance between transmitting nodes (lower transmission power), in order to compensate for greater energy demand on each node.

C. COOPERATIVE DIVERSITY

Cooperative Diversity is a technique used within wireless data networks where multiple quasi-simultaneously received transmission signals can be combined, in order to increase the accuracy of a signal that may otherwise be unreadable due to signal attenuation or interference between the separate signals. Since the previous thesis work being used as a comparison to the simulations created in this thesis utilized a wireless network device that implemented some form of this capability, we have included a brief explanation of this relatively new wireless networking capability.

1. How Cooperative Diversity Works

According to conventional wireless networking schemes, if a wireless receiver detects multiple signals being transmitted across the same channel, the node may have to ignore what it receives and consider the combination of both transmissions to be simple noise on the channel, because it is unable to distinguish one signal from the other. Using cooperative diversity, a wireless receiver can recognize multiple occurrences of the same transmission, even if they are slightly out of synch with each other, and combine both occurrences into a single amplified signal. Essentially, this capability mimics the use of multipath signal combinations, or special diversity, where a node receives several signals overlapping in time that share the same source. These signals arrive at slightly different times due to the difference in path lengths caused by reflections or refractions of the signals. The key difference with cooperative diversity is that the multiple received signals are actually received from different transmission sources (nodes) that each received and relayed the same source packet, without modification (save for identical time-to-live changes and resulting integrity check (CRC) modifications, if used) resulting in the multiple receptions. The implementation of this capability, generically modeled by [1], is patented by TrellisWare, Inc., which provides a more detailed description of their technology in [4]. The general concept is described below

As shown in Figure 4, when S transmits a signal, the signal can be received by any number of R_n stations. If more than one of the R_i station attempts to relay the signal to D, then D will receive multiple instances of the same signal originated by S. Since the

distance between each R_i station and both S and D are slightly different, each of the Ri_n station's transmission will arrive at D at slightly different times. Multiple versions of the same signal can occur in networks where multiple nodes act as transmission relays for another node, or if a single node's signal is partially reflected off of a physical terrain object and redirected in the same direction as the signal's intended receiver.

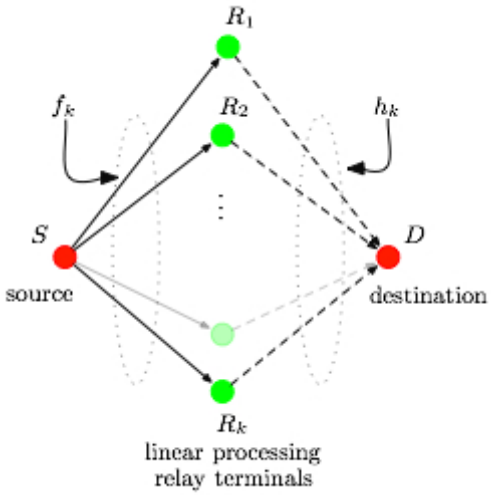

Figure 4. Cooperative Diversity Model (From Swiss Federal Institute of Technology Zurich website).

2. Why Cooperative Diversity Is Useful

Cooperative diversity can be particularly useful for networked nodes that are operating within a highly reflective (i.e., urban or extremely rocky) environment. The wireless node evaluated in [1] implements a version of cooperative diversity, in an attempt to increase the reliability of each transmitted signal and to alleviate the potential confusion caused by a single node's reception of multiple identical signals from separate nodes retransmitting identical signals from a more distant signal originator. Since each of these nodes will relay a transmission for which they are not the intended recipient, there is a high potential for multiple non-destination nodes to receive a transmission at approximately similar times, and then retransmit them such that identical signals arrive at the distant destination node at close to identical times. If there was no type of

cooperative diversity implemented at the receiving node, then in any network that consisted of more than two nodes, it would be very likely that most received transmissions would be regarded as interference and discarded, because of the difference in the timing of the multiple relayed signals.

While data networks in general are extremely complicated entities, mobile wireless networks introduce additional complexities that further complicate the deployment of effective wireless data communications. Understanding the different levels of network functionality is crucial to both understanding the problems encountered with these networks and the benefits and limitations to proposed solutions to each problem.

The following chapter discusses three mobile wireless network devices. The first two are currently deployed wireless communication devices, one of which provides only simple mobile single-hop wireless data links and the other provides an early version of a mobile multi-hop wireless data link network. The third device is not currently deployed, but it uses some emerging mobile wireless communication technologies, including cooperative diversity, to provide enhanced mobile wireless communications that could greatly increase the tactical communication capabilities over currently deployed devices.

THIS PAGE INTENTIONALLY LEFT BLANK

III. ANALYSIS OF DEVICES DISCUSSED IN SIMULATIONS

A. SINCGARS

Since its initial fielding in 1993, the Single Channel Ground and Airborne Radio System (SINCGARS) has been the primary tactical communications device of the US Military. While three of its six radio terminal versions support data communication links, this radio was designed and deployed primarily as a voice communications device.

The SINCGARS radio is worth mentioning in this thesis because of its prominent presence within all levels of tactical command and control architectures. This radio provides both data and voice communication services, but its low data rate limitations essentially disqualify it from being a candidate for running the same suites of data network applications as the EPLRS radio (described later in this chapter) or as more data-capable emerging mobile wireless networking devices. Its general data characteristics and network simulation performances are included here for completeness, and because these capabilities are not entirely without benefits.

The SINCGARS radio generates frequency modulated (FM) signals within the low VHF frequency range of 30-87.975 MHz, and can be used in either single frequency or frequency hopping modes. The data capable versions of the radio terminal can transmit in five different data modes, ranging from 600 bps to a maximum data rate of 16 Kbps. Its data transmissions are limited to a single FM analog data stream, and the radio terminal has no built-in error correction capabilities. Since this type of data signal is very susceptible to signal interference, when a SINCGARS radio terminal is transmitting at its maximum data rate of 16 Kbps, its line-of-sight transmission planning range is cut almost in half, as indicated in Table 1. This is because the higher throughput analog signal is more susceptible to noise introduced into the transmission at the higher power levels required to travel the longer distances. The SINCGARS does not include any kind of embedded MAC protocols, so any network access control must be controlled at the application layer. Since the SINCARS is typically a single-hop communication device,

this would severely restrict the tactical commander's ability to maneuver units in a manner that ensured continuous data connectivity via SINCGARS data links.

TYPE RADIO	DATA RATE	RF PWR	PLANNING RANGES*
MANPACK/ VEHICULAR	600 - 4800 bps 16000 bps	HI HI	3 km - 5 km 1 km - 3 km
VEHICULAR ONLY	600 - 2400 bps 4800 bps 16000 bps	PA PA PA	5 km - 25 km 5 km - 22 km 3 km - 10 km

* **Vehicular radios only.**

Note: Planning ranges are based upon line of sight and are average for normal conditions. Ranges depends on location, sighting, weather, and surrounding noise level, among other factors. Use of OE-254 antenna will increase ranges for both voice and data transmissions. Enemy jamming and mutual interference conditions will degrade these ranges. In data transmissions, use of lower data rates will increase range.

Table 1. SINCGARS Data Transmission Maximum Planning Range (From [11]).

Even if a retransmission site were used to support these links, similarly to how analog voice communication ranges are extended in SINCGARS networks, it would still not help extend these ranges. Since SINCGARS retransmissions are accomplished by connecting two SINCGARS radios by a cable and having one radio automatically relay any signal received by the other, the retransmitted signal is simply an amplified version of the original analog signal. Since the radio does not process and regenerate the data packet being relayed, the retransmitted signal would still contain any degradation, such as interference from other signals or noise introduced by environmental factors, which keep the data from being understood by its destination.

Despite these range limitations, point-to-point data applications using the SINCGARS have existed for some time. Since there are no data generating applications organic to the SINCGARS radio terminal, the data transmitted across a SINCGARS data link must be generated by an externally connected device. Also, because there is no routing capability within the device, data links are typically limited to one-hop communications. Some specialized data generation devices take advantage of the data capability of this radio, but only allow single-hop data communications between SINCGARS radios and limit data traffic to simple, preformatted text messages.

A regular personal computer (PC) can be connected to the SINCGARS radio terminal via a specialized data cable, which connects the radio terminal's data port to the computer's serial port. In this configuration, each data transmission can only be received by a similarly configured SINCGARS radio terminal with an attached PC. There is no traffic limitation to the types of files sent across such a link, but because of the link's small throughput (16 Kbps), it would not be well suited for applications with heavy traffic requirements. The requirement of additional data equipment also places increased weight and physical space restrictions on users wishing to transmit data using SINCGARS radios.

Since the SINCGARS radio was designed as a voice-centric physical layer communication device, it does not contain any preprogrammed data routing or MAC protocols for the use in creating multi-node data networks. As a result, it is up to the attached data device to perform data routing and MAC functions for the network.

As explored in [6], it is shown that it is possible to implement more modern types of data networks, such as mobile ad hoc networks (MANETS) between multiple devices, using the SINCGARS radio terminal as the physical layer device. This was accomplished using the data cable to connect a PC to the SINCGARS radio terminal, and running software implemented link and network layers that receive transmitted data packets and either accepts them, passing them to the appropriate application or retransmits them to other SINCGARS radios.

This application allows the radio to retransmit a newly created version of the data packet (without any noise that may have interfered with the initial signal), instead of merely relaying an amplified version of the same signal. This method of packet retransmission could overcome some of the range limitations for retransmitted data signals in a SINCGARS network, by eliminating any noise that may have been introduced to the original analog signal.

Another development of technologies available for using SINCGARS radios for data traffic is the Internet Controller (INC) circuit card that can be installed in the vehicle amplifier adapters (VAA) for the newer versions of SINCGARS radios. This device acts

as a basic IP router that allows for the routing of data to other SINCGARS radios, EPLRS radios and other PCs connected to the INC. It allows an external data device to be connected via an included Ethernet port and it is compatible with any device that supports 802.2 and 802.3 protocols. It also provides improved forward error correction (FEC) capabilities and claims to increase the planning range for its data links through digital conversion and retransmission. However, the INC is not dismountable, so it does not provide data capabilities to foot mobile troops, and it does not increase the throughput of a SINCGARS data link. In fact, where it is implemented, the overhead required for FEC will actually reduce the amount of usable data throughput.

Regardless of the mode of data transmission across a SINCGARS data link, the data rate limitations of such implementations still restrict communications to data traffic with very small throughput requirements, such as sending simple text messages and would not provide sufficiently large transmission channels for the use of VoIP services or the exchange of larger data files.

B. EPLRS

The Enhanced Position Location Reporting System (EPLRS), which was initially fielded in 1972, was an alternative to satellite-based position tracking of friendly ground forces. It did this by providing a secure IP-based data backbone that networked digital position information between each EPLRS radio operating across the same network. Since then its role has shifted from just a position tracking device to enabling limited types of general mobile wireless data networking for tactical commanders. That is, it provided a platform for hosting applications other than position tracking. Unlike the SINCGARS, it does not provide integrated voice communications capabilities, but the data throughput is significantly greater than that of the SINCGARS and is high enough to support externally connected voice over Internet Protocol (VoIP) technologies.

The EPLRS radio system operates within the UHF spectrum at between 420 and 450 MHz and is capable of providing multiple devices simultaneous access to different types of data channels (different channels may use different MAC protocols), through its use of both Time Division Multiple Access (TDMA) and Frequency Division Multiple

Access (FDMA) resource sharing. The maximum aggregate data throughput is between 525 and 1000 Kbps, depending on the version of EPLRS terminal being used. A regular PC can be connected via an Ethernet port that is built-in to the EPLRS radio device, which is a much less restrictive interface than the serial-to-data port PC connection required for a SINCGARS radio terminal.

The EPLRS radio device establishes its data links by creating a series of permanent virtual circuits (PVC) between different EPLRS devices, called *needlines*. A needline can be either many-to-many, few-to-many or one-to-one, and each EPLRS radio device can simultaneously communicate across 28 different needlines. Figure 5 shows a conceptual illustration of how EPLRS network resources are organized and divided into multiple needlines.

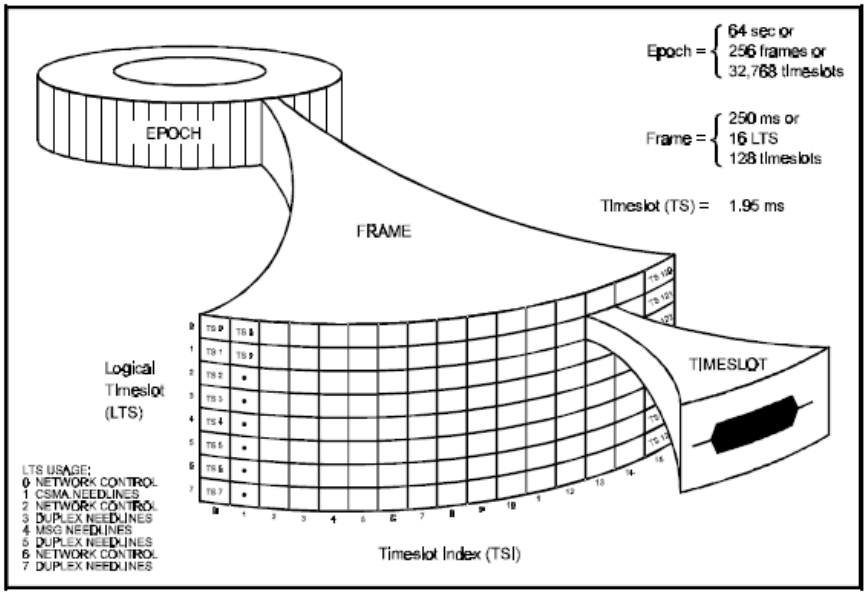

Figure 5. EPLRS Epochs, Frames and Timeslots (From [12]).

The smallest division within a needline is the *time slot*. Each EPLRS radio device assigned to a particular needline can be assigned one or more time slots within the needline. These timeslots can be either 2 ms or 4ms in size, and time slot size must be consistent across the entire network (i.e., different needlines cannot operate with different time slot sizes). Time slots are then grouped into *frames*, which consists of 128 consecutive time slots. Groups of timeslots within each frame are combined into eight

different *logical timeslots* (LTS) that span across multiple frames and can each be assigned to different needlines. Frames are further grouped into *epochs*, which are 256 consecutive frames, which is the largest time division used in EPLRS. This multi-tiered method of organizing network resources allows the network manager greater flexibility and control in allocating network resources to accommodate the needs of many different types of users. However, this can also restrict the amount of resources available to any one needline, as once a resource is assigned to one needline it cannot be used by another. Each radio can support up to 32 separate needlines (4 of which are typically used for network control), which can be transmitted across any one of up to 8 frequency-separated channels.

Each EPLRS network is configured through the EPLRS Network Management (ENM) suite, software that is hosted by a computer connected to an EPLRS radio. The ENM allows the network manager to create needlines, assign network resources to these needlines and set the MAC protocols for each needline.

The EPLRS radio supports many different MAC protocols. Since each needline operates on a separate logical channel from all the other needlines, each needline can be assigned a different MAC protocol, according to the needs of the user. There are five main types of MAC protocols available to the EPLRS radio system: Carrier Sense Multiple Access (CSMA), Multi Source Group (MSG), High Data Rate (HDR) Duplex, Low Data Rate (LDR) Duplex and Dynamically Assigned PVC (DAP).

A CSMA needline allows many-to-many transmission capabilities across the needline at data rates between 150 bps and 485.76 Kbps. CSMA needlines create one-time communication paths between two (or more) EPLRS radios, and transmissions sent via a CSMA needline are not acknowledged by the receiving EPLRS radios. Even if there are multiple transmissions that must travel between the same sets of nodes, since EPLRS nodes are typically highly mobile, these transmissions may not always be able to travel along the same path, so the path for each transmission (i.e., the nodes that relay messages across multi-hop links) will be recalculated for each transmission. Network resources are assigned to radios on-demand, based on availability. While there is no guarantee of resource availability to any radio on the network, a maximum hold time

value can be set that prevents any one device from holding the needline resources for longer than the determined maximum hold time. This ensures that one radio cannot prevent other radios from utilizing the needline, by permanently capturing the CSMA needline resources.

A CSMA needline can support a maximum of 6 hops between originating and destination radios, but network managers can limit the allowed number of hops to 2 or 4, in order to reduce wasted bandwidth. Lowering the maximum hop count reduces wasted bandwidth, because the EPLRS implementation of this type of needline requires each traffic originator to wait *n*-1 time slots between transmissions, where *n* is the maximum hop limit, in order to ensure that the relay transmissions of tightly clustered nodes cannot interfere with the introduction of new messages. All EPLRS radios do not act as a relay for all messages. Instead, the relay nodes involved in each transmission path are chosen by a proprietary relay assignment algorithm that is processed by each receiving radio.

MSG needlines provide few-to-many transmission capabilities across an EPLRS network that guarantees bandwidth for up to 16 simultaneously transmitting radios, at a data rate between 37.5 bps and 485.7 Kbps. MSG needlines create one-time communication paths between EPLRS radios, where traffic paths between nodes are recalculated for every transmission, and transmissions sent via a MSG needline are not acknowledged by the receiving EPLRS radios. Specific time slots can be reserved within a MSG needline for specific radios, regardless of the radio's actual need for them, which is not possible using a CSMA needline. Also, transmitting radios must only wait for a single time slot between its transmissions, resulting in less wasted bandwidth than the CSMA needline.

LDR Duplex needlines provide one-to-one data links between two EPLRS radios at data rates between 20 bps and 16.2 Kbps, which is very similar to the transmission rates of the SINCGARS radio link. LDR needlines create two-way communication paths between two radios, and each transmission across this type of needline is acknowledged by the destination radio. Resources used for all LDR needlines on a network are reserved prior to the deployment of the network, and are allocated to specific LDR needlines on-demand, based on availability. If all of the LDR resources for a particular network are in

use, then additional LDR needline transmissions will be delayed until resources are released by other LDR Duplex needlines. Similarly to MSG needlines, each relaying radio must only wait for a single time slot between its transmissions, in order to minimize wasted bandwidth.

HDR Duplex needlines are very similar to LDR Duplex needlines. All of the HDR resource assignment must occur prior to network deployment, and will only be changed if an ENM adjusts these assignments during network deployment. The most significant differences are that the data rates for these needlines range from 600 bps and 242.9 Kbps, and network resources are reserved for each individual HDR needline. So, since all HDR needlines are not pulling from the same pool of bandwidth, there will always be network resources available for each HDR needline, eliminating the delays that could be experienced by LDR Duplex needline transmissions. The disadvantage of this is that while there are HDR needlines not actively sending data, the resources allocated for these needlines are being essentially wasted.

DAP needlines are simply either LDR or HDR Duplex needlines that are implemented on an on-demand basis. Unlike LDR and HDR needlines, they are not maintained throughout the operation of the network and are torn down once they are not needed.

While the currently deployed EPLRS radio does provide a fair amount of versatility for the configuration of relatively efficient data networks, its large size and the high power consumption of the system prevent it from ever being a viable candidate for providing highly mobile data connectivity to foot mobile troops.

C. COOPERATIVE DIVERSITY RADIO

The primary purpose of this thesis is to compare the network performance of existing ground-based tactical radio technologies to that of a specific radio device that uses emerging mobile wireless technologies, which was modeled in previous thesis work. Within this thesis, this radio will be referred to as the Cooperative Diversity Radio (CDR). A very brief description of the CDR is given here, but more detailed descriptions can be found in [1].

The CDR is primarily different from currently available technologies in that it uses cooperative diversity, spatial slot reuse and spatial pruning of data flow to perform multi-hop network flooding of a single wireless transmission significantly faster than any other known wireless technologies. It is designed to provide both voice and two-way data communication services from the same handheld device. An external data device is still needed to create data for transmission across a CDR network, that is, the device does not include any application hosting capability. It does, however, provide an 802.2 and 802.3 (Ethernet) interface to connect the device to a wired network, in which case it serves as a bridge between the wireless and wired domains, but as it does not host network protocols, it does not serve as a router, per se. The specific MAC protocol used for the CDR is not known, but it was previously modeled using the slotted-Aloha MAC protocol, since this protocol was determined to provide the most meaningful observations regarding the performance of such a device.

In a CDR network, every radio can act as a potential relay for another radio's transmission. This does not mean that all radios always relay all transmissions. A proprietary algorithm is used to ensure only the radios that are likely to be on the path between source and destination are used as relays during the creation of a two-way link between two nodes. The proprietary implementation transmits within the UHF frequency spectrum and has a maximum hop count of 9 hops within a CDR multi-hop network, driven by the maximum delay tolerance of the multi-hop voice communications supported by the radios, a capability unavailable for either the EPLRS or SINCGARS.

Now that we have discussed the general characteristics of a variety of mobile wireless networking devices, we will now explain our methods for their evaluation. Since we do not have the resources available to setup networks involving actual versions of each of these radios, we will instead analyze an approximation of their performance generated through the use of network simulation software.

The following chapter discusses how the characteristics of these devices will be translated into software models, which will be used by software network simulation programs to analyze and compare the performance of each of these networks, under a variety of network loads.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. JCSS IMPLEMENTATION

A. JCSS NETWORK SIMULATION SOFTWARE

The Joint Communications Simulations System (JCSS) is a network simulation software suite that has been adopted by the Joint Chief of Staff J6 directorate as the primary military network modeling and simulation tool for the Department of Defense. Formerly known as NETWARS, it is maintained by the Defense Information Systems Agency (DISA), and provides the Department of Defense with a standardized tool for the analysis of the behavior and performance capabilities of available defense communication networks. Its main goal is to allow military communication and acquisition planners to identify and quantify potential risks and deficiencies that may exist within network configurations prior to their deployment and to validate any proposed hardware or configuration changes made to existing systems. JCSS targets users ranging from lower level operational planners, who are attempting to quickly verify an equipment density list for a specific operation, to higher level analysts, who are attempting to estimate network load limits, identify bottlenecked links, or analyze the performance and interoperability of different software configurations across worldwide networks.

JCSS can be generally divided into two major components: the network configuration interface and the capacity-planning interface. In the network configuration interface, the user can choose from a menu of preconfigured communication nodes (or import custom nodes) and "drag-and-drop" them into the software's workspace. The workspace can be a generic flat area of various dimensions, or it can be made to simulate a very specific real-life geographic location, incorporating map information, satellite imagery, and digital terrain and elevation data, in order to increase a simulation's ability to reflect network performance within a specific geographic area.

Once all of the communication nodes are in place, each of them can be connected to each other with different types of communication links, as appropriate to the specific devices that define how each device will interact with the others. Also, each node has a

list of attributes that can be adjusted to reflect the capabilities and configurations available to the actual communications system being modeled.

Once the simulated network is fully configured, the user can begin using the capacity-planning interface to start their analysis of the performance of various types and quantities of network traffic. In order to accurately generate network traffic across the simulated network and collect statistics related to this traffic, JCSS operates in conjunction with a commercially available OPNET Modeler network simulation suite, licensed from OPNET Technology, Inc., which is described later in this chapter. Since this simulation engine contains many features that are not applicable to the configuration of military network models, JCSS attempts to streamline the OPNET Modeler simulation engine's interface to make it more familiar to analysts who are attempting to model military-specific communication networks.

As with any simulation model, the JCSS software suite does not provide an exact replication of radio performance within real-world scenarios. It does, however, provide a reasonably accurate frame of reference for the comparison of the advantages and disadvantages of employing various types of networked devices under many different network conditions.

1. OPNET Modeler Network Simulation Suite

The OPNET Modeler network simulation suite was designed to model the performance of many different types of commercial data and voice communication devices and to provide the user with tools to easily analyze the performance of these devices under a wide variety of configurations. OPNET Modeler also allows for the creation and analysis of customized communication device models and model processes, which means that the user is not limited to the use of the preconfigured models that come packaged with the software. Since all of the JCSS network simulation capabilities are derived through its use of OPNET Modeler device node models and its discrete event simulation kernel, it is important to understand how the OPNET Modeler software functions, and how it is able to model specific types of tactical radio nodes. The rest of this chapter includes a summary of the more specific descriptions of the functionality and

capabilities of OPNET Modeler compiled in [5]. All of the simulations performed in this thesis were run using OPNET Modeler version 14.5.

2. Node Models

Since the objective of most modeling efforts is to observe the behavior and measure the overall performance of networks consisting of many nodes, it is critically important that the model of each node included in the network is sufficiently representative of the actual device being modeled, as well as an appropriate modeling of the expected traffic types and volumes. It is important for the OPNET Modeler user to be familiar with the general node model design methods used to create the models of the devices included in their simulation.

A node model is a software representation of an actual network device. Each node model contains its own internal processes that operate independently of the processes contained by other node models within the same network. Each node model can execute multiple processes and can contain multiple types of interfaces between other nodes. There are different characteristic variables for each node model that limit its behavior to actions that mimic those performed by the device being modeled. A model's performance characteristics can be adjusted by altering the available configuration settings for each model. Each model configuration setting is designed to reflect the types of settings available on the actual device being modeled, but many models include settings that are not found on the actual device, in order to make it easier for the user to quickly configure a simulation where specific configuration settings are not important to the analysis being performed.

The actual functionality of each node model is defined by a finite-state-machine set of process models that determine which of its processes are called at different simulation times and in response to different received signals from other nodes. Each state transition in the various process models is defined by blocks of C or C++ code that control the behavior of each process, and the order in which each state is reached as the simulation progresses.

Each process model can be altered at the code level, in order to give the user complete control over the functionality of customized communication nodes. This ability to allow detailed customization of various types of node models gives an OPNET Modeler user the ability to simulate the performance of nodes that do not exist in real life, or for which there is no ready-made model provided.

3. Discrete Event Simulation Kernel

The discrete event simulation kernel is the "brains" of each network simulation. It orchestrates the generation and processing of all events that occur during a simulation. While there are thirteen specific types of events recognized by the simulation kernel, the definitions of each specific event is well beyond the scope of this particular thesis. In general, an event is one of two types of actions: It is either an instance of one node sending network traffic to another node, or an instance of a process within a single node communicating with a different process of the same node, such as the encapsulation of one protocol data unit into another. When any node requires an action to be performed, it sends an interrupt to the discrete event simulation kernel, which is queued and processed according to the needs of all other nodes within the same network. Since only one event can have access to the kernel resources at a time, it is the discrete event simulation kernel's job to ensure that the resulting behavior of any portion of the network simulation is minimally affected by the order in which each event is processed. In order to accomplish this, the discrete event simulation kernel must be able to process events serially in a manner that makes it appear as though they were processed in parallel. This is because each simulation will likely be modeling multiple communication devices that would normally be functioning simultaneously in a non-simulated environment. However, by dividing a continuous timeline into sufficiently small discrete intervals, each interval can be established such that only one event occurs during that interval.

In order to achieve effective event processing, the discrete event simulation kernel uses an event list to manage each event. Instead of using a simple FIFO queuing method, where each generated event is queued based on when it was processed by the kernel, each kernel event has a simulation-time timestamp and is placed in a proprietary data structure

(no specifics of the exact functionality of this data structure were given in the OPNET Modeler documentation) based on each event's simulation-time timestamp. So, if an event is generated with a simulation-time that is earlier than that of the event currently at the head of the data structure, the new event will be scheduled for processing before all other currently scheduled events. This essentially makes use of a type of priority queue data structure.

4. Link Models

OPNET Modeler has the ability to model many different types of physical layer links between networked nodes. The included link models range from Ethernet and fiber optic cables to simple single channel radio wave transmissions and complicated frequency hopping transmissions. Since the only physical layer link used for simulations in this thesis is wireless radio wave transmissions, this section will focus on how OPNET Modeler represents radio wave transmissions in its software.

Each link simulated in OPNET Modeler is represented by a series of pipeline stages that simulate the physical effects of the transmission medium on the transmitted signal. Pipeline stages are implemented as procedures written in C or C++ programming languages. Each procedure typically takes a simulation-packet data type as their only argument, and returns a simulation-packet data type to the wireless receiver of a networked node.

For wired links, the pipeline effect on each transmission can be computed once at the beginning of the simulation and the same pipeline can be used throughout the entire simulation, as the characteristics of a wired transmission medium remain static. Since there are so many variables that can have a significant effect on wireless transmissions, wireless links must be evaluated dynamically at each simulated packet broadcast. At each signal broadcast, a series of virtual links are created between the transmitting node and any wireless receivers within range of the transmission. Once each virtual link is established, a copy of the transmitted packet is created for each link and a pipeline analysis is performed on each individual copy of the transmitted packet. Each stage of the pipeline analysis takes into consideration a different environmental or signal factor

that would have a significant effect on an actual radio wave transmission. Since no actual radio waves are transmitted, the analysis results are found mathematically, using characteristics of each simulated transmission (i.e., power level of the transmitted signal and physical distance between the transmitter and receiver) and formulas involving known physical characteristics of the real world (i.e., the propagation delay *d* of a signal travelling distance *r* will be $d = r/C$, where *C* is the speed of light). For the Radio transceiver Pipeline, there are fourteen stages, each of which is briefly described below:

a. Receiver Group

This stage is executed one time at the beginning of the simulation. The purpose of this stage is to determine which nodes are likely to communicate with each other, based on link type and relative position, in order to prevent the remaining pipeline stages from unnecessarily analyzing the effects of nodes that are of a different type or are currently too far away to have any effect on the link being analyzed. This stage has no influence on the performance or behavior of the network being simulated, and is only included to minimized the amount of total time it take for the simulation to run, by eliminating unnecessary calculations.

b. Transmission Delay

Based on the packet size and transmission rate of the transmitting node model, this stage calculates the amount of total time that passes from when the node begins to transmit its signal to when the node completes its transmission. This is information will be used to calculate the final results of later pipeline stages and is executed once per transmission.

c. Link Closure

Since the Receiver Group stage analysis is only performed at the very beginning of the simulation, this stage accounts for potential changes to the topology of the wireless nodes in the network that have occurred since the simulation began, and adjusts the list of nodes to be considered for signal interference processing. This stage is

performed once per receiving channel and has no influence on the performance or behavior of the network being simulated. It is only included to minimize the amount of total time it takes for the simulation to run, by eliminating unnecessary calculations.

d. Channel Match

This stage simply compares the channel frequency settings of both nodes on each virtual link. If the frequencies of both nodes on a particular virtual link match or are close enough to potentially interfere with each other, then the transmission will be forwarded to the next stage of the pipeline analysis. If they do not match, then the virtual link between the two nodes is no longer considered for further pipeline analysis of the current transmission.

e. Transmitter Antenna Gain

During this stage, the intensity of the transmitter's signal in the direction of the receiving node is calculated and used later in the pipeline analysis, in order to determine if the transmission's strength is great enough to reach the receiving node. This variable is of special significance to nodes transmitting with directional antennas.

f. Propagation Delay

This stage calculates the simulation-time at which both the leading and trailing edge of the transmission actually reach the receiving node. This information is used later in the pipeline analysis to determine if any other signals arrive at the receiving node during the reception of the transmission being analyzed.

g. Receiver Antenna Gain

During this stage, any increase in intensity of the signal received by the receiving node from the use of directional antennas is calculated and used later in the pipeline analysis.

h. Received Power

This stage calculates the actual power of the signal received based on the transmission power, antenna gains and node distance, which were previously calculated by earlier stages of the analysis. This information is ultimately used to determine if the transmission's strength is great enough to overcome any interference from any other transmissions received at the same simulation time.

i. Interference Noise

This stage calculates the noise created by other transmissions, based on the power of each interfering signal and the duration of each interfering signal.

j. Background Noise

This stage calculates the noise created by other sources of electromagnetic radiation that have been introduced into a simulation. These values are based on the type of background noise and its proximity to the receiving node's antenna.

k. Signal-To-Noise Ratio

This stage calculates the signal-to-noise ratio for each part of the transmission, using the received power, interference noise and background noise values determined in earlier pipeline stages.

l. Bit Error Rate

This stage determines the expected bit error rate of the transmission based on the signal-to-noise ratio calculated across the entire transmitted frame. Greater amounts of noise will produce a higher probability of erroneous bits.

m. Error Allocation

It is at this stage that actual errors are introduced into different parts of the transmitted frame, based on the bit error rate probability values determined for different sections of the transmission.

n. Error Correction

No error correction algorithms are actually performed during this stage of analysis. Instead, this stage simply decides whether or not to discard the transmission based on calculations that are performed using probability distributions based on the specific error correction mechanisms being used in the network and the total number and locations of the introduced bit errors.

5. Application Profiles

In order to consistently simulate many different types of network traffic across a network with many different nodes, OPNET Modeler allows its users to use application profiles that mimic various combinations of network traffic that may be produced by each node. This makes configuring multiple instances of the same type of user quicker and more consistent. If adjustments need to be made to an application profile prior to the execution of a simulation, once changes are made to the profile they will automatically be reflected on all nodes assigned to that profile. This is especially useful when running different variations of the same general simulation scenarios.

While some preconfigured application profiles that simulate common types of network activity (internet browsing, email, ftp, video conferencing, etc.) are provided, it is also possible to create and configure custom applications that generate the specific types of traffic across the network. Different application profiles can be assigned to different nodes, and a single node can be assigned multiple profiles. These application profiles can also control the frequency and destination of defined traffic flows.

Since the mode and tempo of information flow in military communications can vary significantly from that of a typical business office network, the custom application options were best suited to modeling data traffic of tactical data networks in our simulations.

6. Data Collection

OPNET Modeler allows the user to collect data on any kind of event generated during the simulation. Statistics can be collected for the network as a whole, or for any single link, node, or application operating within the simulated network. The user simply selects the types of data to be collected prior to running the simulation, and then after the simulation is complete, the collected data can either be displayed via an included graphical interface, or exported to separate programs for analysis.

B. JCSS SINCGARS MODEL

In our network simulations, we will use the SINCGARS node model included with the JCSS version 8.0. We have made no alterations to this model and, in order to establish a basis for comparison to the CDR node, will attempt to introduce the same type of network traffic using the SINCGARS model as the network's physical layer transmission device. This section briefly explains how the SINCGARS radio is modeled in JCSS.

1. Node Model Logic

In order to accurately model the functionality of the SINCGARS radio, JCSS software provides a device model that attempts to replicate the logic behind the flow of data through a SINCGARS radio terminal. As depicted in Figure 6, the internal organization of the node model divides all of the separate functions into individual process models (the gray squares), which are able to communicate with other processes, as indicated by the red and blue arrows.

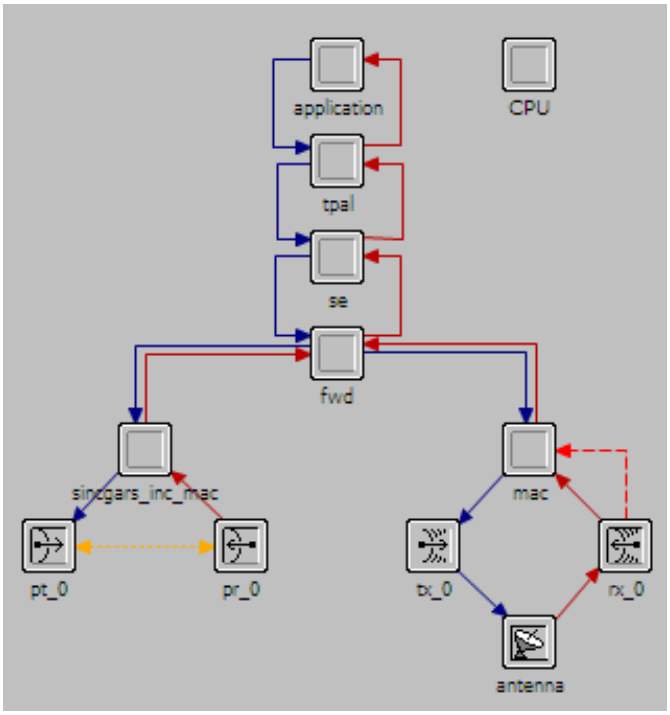

Figure 6. SINCGARS Node Model

The Application process model (*application)* represents the applications available to the SINCGARS radio, which consist solely of analog sound wave input and output via the radio handset. The Transport-to-Application process model (*tpal*), which is present in most OPNET device models, provides an interface between the application and transport protocols available to each device model. This simply allows each application model to be reused by multiple types of devices. The Transport Protocol process model (*se*) represents the only transport protocol available to this model, and the Packet Forwarding process model (*fwd*) merely determines the proper forwarding direction for each packet it encounters (analog voice is sent to the *se* model and analog data is sent to the *sincgars_inc_mac* process). The SINCGARS Data Interface process model (*sincgars_inc_mac*) represents the interface between the radio terminal and the externally connected Internet Controller (INC). The Media Access Control process model (*mac*) simply models the interface between the radio and its antenna.

2. Node Model Configuration

For data communications, we will also incorporate the SINCGARS INC interface model into each node. The INC functions as a basic IP router and allows a normal computer to connect to the SINCGARS radio via an Ethernet port. OPNET Modeler provides SINCGARS INC interface model (*sincgars_inc_adv*), shown in Figure 7, that provides the logic behind the data routing and MAC protocol capabilities of the SINCGARS INC module.

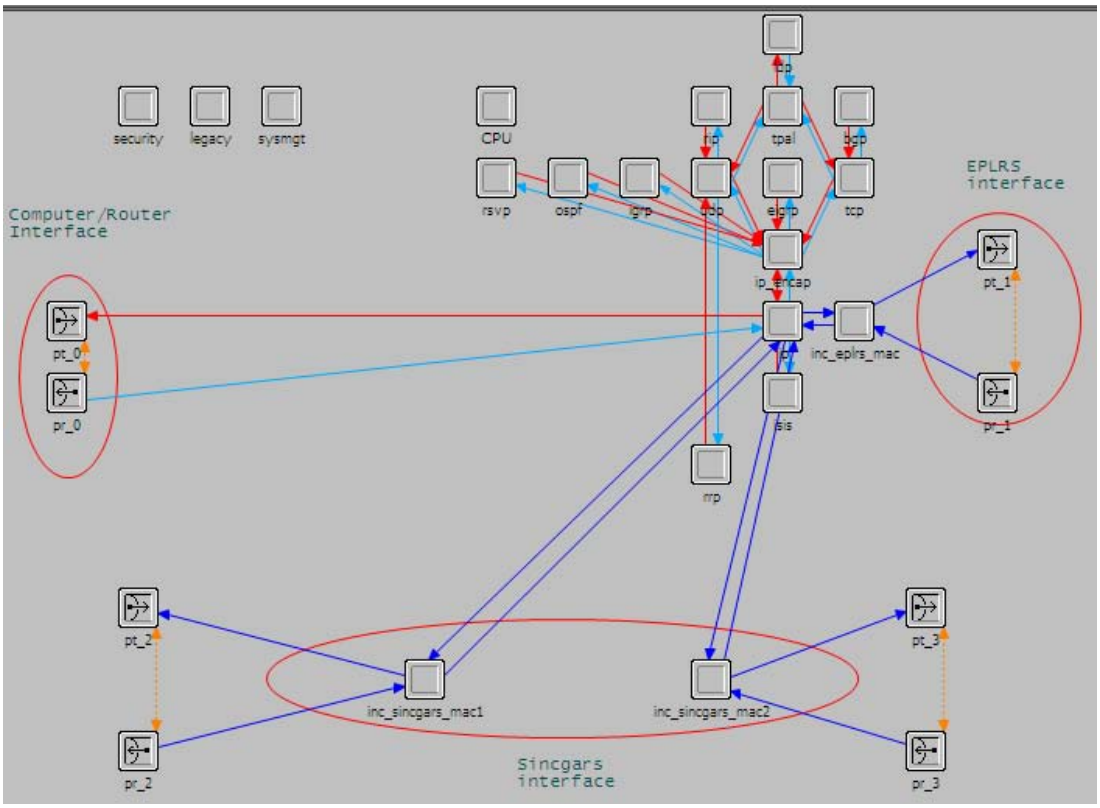

Figure 7. SINCGARS Internet Controller Model

Notice that there is no *application* process module included in the SINCGARS INC interface model, like there is in the SINCGARS radio terminal model. This is because there are no voice communication functionalities to model the INC, as all voice communications are not handled by the INC. Additionally, any applications that would be producing data traffic for transmission through the INC would be hosted on an externally connected data device. Since the only task performed by the INC is deciding

how to route various types of data transmissions, the application process is replaced by a Label Distribution Protocol process model (*ldp*), which builds and maintains a database of its peer nodes for transmission relaying purposes. The Transport-to-Application process model (*tpal*), simply acts as an interface between the ldp and udp and tcp processes, so the same ldp process can be reused by other device models. Both the Transmission Control Protocol (*tcp*) and User Datagram Protocol (*udp*) process models are included, since the INC supports both types of transmissions. The IP encapsulation process model (*ip_encap*) performs the packet encapsulation functionality for the Internet Protocol process model (*ip*), which focuses on the forwarding required to run the IP protocol. The EPLRS MAC (*inc_eplrs_mac*) and SINCGARS (*inc_sincgars_mac1/2*) process models control both incoming and outgoing packets' access to the INC Ethernet port, both SINCGARS data ports and the EPLRS data port.

C. JCSS EPLRS MODEL

In our network simulations, we will use the EPLRS node model included with the JCSS version 8.0. We have made no alterations to this model and have decided to use CSMA needlines as the MAC protocol, since it is the only many-to-many capable needline currently supported by OPNET Modeler. This section briefly explains how the EPLRS radio is modeled in JCSS.

1. Node Model Logic

In order to accurately model the functionality of the EPLRS radio, JCSS software provides a device model that attempts to replicate the logic behind the flow of data through an EPLRS radio terminal. As depicted in Figure 8, the internal organization of the node model divides all of the separate functions into individual process models (the gray squares), which are able to communicate with other processes, as indicated by the red and blue arrows.

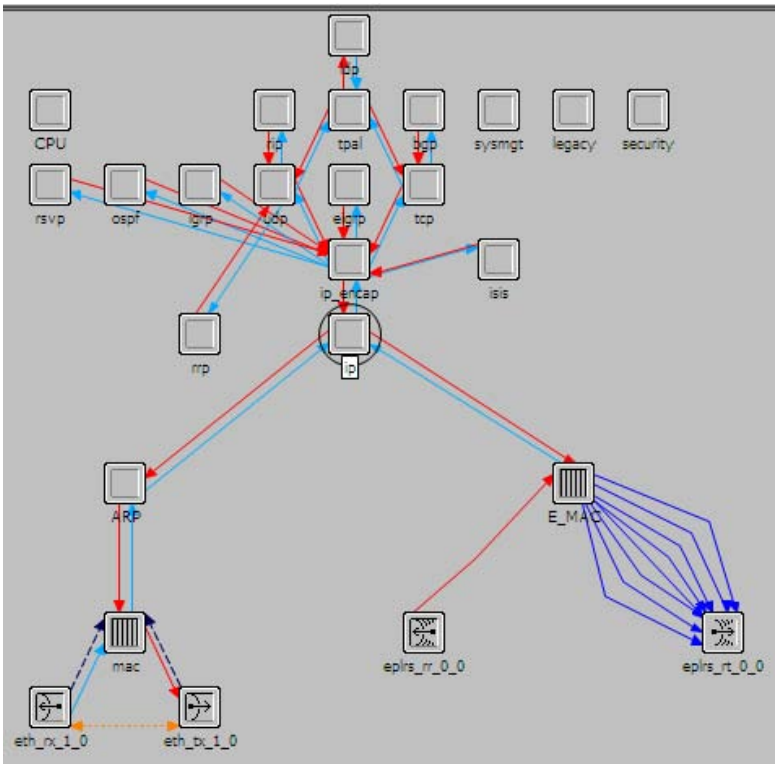

Figure 8. EPLRS Node Model

Similar to the SINCGARS INC model, there is no *application* process module included in the EPLRS node model. This is because there are no voice communication functionalities to model for this radio, and any data application requiring use of the EPLRS network will be hosted on an externally connected computer. Since the only task performed by the EPLRS is deciding how to route various types of data transmissions, the application process is replaced by a Label Distribution Protocol process model (*ldp*), which simply builds and maintains a database of its peer nodes for transmission relaying purposes. The Transport-to-Application process model (*tpal*), simply acts as an interface between the ldp and udp and tcp processes, so the same ldp process can be reused by other device models. Both the Transmission Control Protocol (*tcp*) and User Datagram Protocol (*udp*) process models are included, since the ELRS supports both types of transmissions. The IP encapsulation process model (*ip_encap*) performs the packet encapsulation functionality for the Internet Protocol process model (*ip*), which focuses on the forwarding required to run the IP protocol. The EPLRS MAC process model (*E_MAC*) simply controls each transmission event's access to the physical transmitter

process model, while accepting incoming transmissions from the physical receiver process model, in a manner that mimics the general functionality of the proprietary EPLRS MAC protocols that route traffic in actual EPLRS networks. The Address Resolution Protocol process model (*ARP*) maintains a queue of outbound packets that still need MAC addresses to be delivered and generates an ARP request and response for these packets. The MAC process model (*mac*) controls both incoming and outgoing packets' access to both EPLRS Ethernet ports.

2. Node Model Configuration

The EPLRS radio only handles the routing and MAC protocol enforcement for data transmissions across the EPLRS network, so an external network interface device needs to be connected to an EPLRS in order to create application data. As a result, a similar INC interface model is utilized for use with the connection of data processing devices to an EPLRS device model.

D. JCSS TACTICAL APPLICATION MODELS

In this section, we will explain the various data application sets that could be expected to be employed across different types of tactical data networks, and how these applications are represented in our JCSS network simulations. For consistency purposes, we will attempt to recreate the tactical applications and profiles used in [1]. A description of each application model and different application profiles are provided in the following sections.

1. Application Model Definitions

The specifications for an application that allows a network node to introduce traffic into its network are defined as application models. The models utilized in our simulations are summarized in Table 2, and explained in more detail below:

Application	Service	Msg Size (Distr)	Destination	Interarrival (Distr)	Success
Unicast to gateway	UDP	50 B (const)	gateway	1 min (const)	3 min
Unicast with ACK	UDP	20-160 B (uni)	unicast	10 min (exp)	10 sec
Constant Multicast	UDP	1066 B (const)	multicast	66.6 msec	90% 3 sec
Bursty Multicast	UDP	67-333 x 40 B (const) @ 33.3/sec	multicast	30 sec (exp)	90% 0.25 sec
IRC from client	TCP	20-512 B (exp)	gateway	10 min (exp)	10 sec
IRC from server	TCP	20-512 B (exp)	client	5 sec (exp)	10 sec
TCP push	TCP	500 B – 1 MB (exp)	gateway	15 min (exp)	2 min
TCP pull	TCP	500 B – 1 MB (exp)	client	15 min (exp)	2 min

Table 2. Summary of Modeled Application Set (From [1])

a. Unicast with ACK: Short Message

This custom application model is designed to mimic small, irregularly transmitted, text-based messages, such as sending a single text message or digital call-for-fire request. Each source node will randomly choose one of its designated destination nodes and send that node a single TCP transmission, ranging in size from 20 bytes to 160 bytes. This traffic will be introduced approximately one time every 1200 simulation-time seconds, with an exponential distribution.

b. Unicast to Gateway: Position Update

This custom application model is designed to mimic small, regularly transmitted data messages from each client node to the network's gateway node, similar to position update transmissions. Any node running this application will send a 50 byte UDP segment to the network's gateway node once every minute.

c. Constant Multicast: Video

This application model takes advantage of the preconfigured application models provided with OPNET Modeler. It allows a network node to pull video-only data from the network gateway node. Because of limitations of the MAC protocol used by the CDR node, the video frame size was reduced to 500 bytes at a rate of 2 frames per

second. Although these traffic characteristics are not representative of normal streaming media, we will keep these same frame size and rate setting for our simulations for accurate comparison purposes to the simulations performed in [1].

d. IRC

This is a custom application model that is designed to mimic the type of traffic load introduced to a network by internet relay chat (IRC) applications. Traffic originates from a network node and is sent to the network's gateway node, which initiates a series of message exchanges that occur between the two nodes. There are approximately 60 send/receive exchanges that occur each time an IRC event takes place, and each node's response is delayed for an average of 5 seconds, in order to simulate message-typing response times.

e. TCP Pull: HTTP

This is a custom application model that mimics the type of traffic introduced by various network nodes requesting HTTP downloads across the network. This application begins with a fixed-size HTTP request of 400 bytes from a network node to the network's gateway with the time between requests varying exponentially with a mean delay of 900 seconds. Each requested page size varies uniformly between 500 bytes and 1 MB.

f. TCP Push: Email

This is a custom application model that roughly mimics the type of traffic that might be introduced by various nodes uploading email messages to the network gateway for delivery. Each email message varies uniformly between 500 bytes and 1 MB and is uploaded at time intervals that vary exponentially with a mean of 900 seconds. This may be an unrealistic rate of typical tactical email usage, as we would expect numerous emails to be sent within only a few minutes of each other, while there is a lull in operations, then have much longer periods of no emails, as the tactical commanders are occupied with other operational tasks. However, for the purposes of a three-hour

simulation, it provides a reasonable reflection of how email-sized traffic may affect the overall network performance in the presence of other running applications. Once the upload is complete, the server acknowledges the upload with a 500-byte message that ends the application event.

2. Application Profile Definitions

Every node within a simulation network will introduce quantities and types of traffic onto its network according to the applications models assigned to it. Since most node users will typically use more than one type of application, applications for our simulations have been grouped into application profiles that correspond to the type of user at each node.

In addition to grouping applications, since it is unrealistic for each node to begin transmitting traffic at precisely the same time (i.e., at the very beginning of the simulation), our application profiles allow additional configuration options that control the simulation start and stop times for executing the application events for each type of user and the frequency that these events are executed. In order to be consistent with the simulations run in [1], we have set all of our application profiles to randomly begin the execution of their application events somewhere between 60 and 600 simulation-time seconds after the beginning of the simulation.

a. Tactical Commander

This application profile models the types of traffic introduced to the network by platoon and company commanders. Due to configuration limitations inherent to OPNET Modeler's implementation of the application profile, the needs of the tactical commander node had to be divided into two separate custom profiles. This is because some of the applications will be run constantly through the entire simulation, and other applications will only run at randomly generated time periods during the simulation. The first profile is assigned the IRC application model, which will be running throughout the entire simulation. The second profile will be assigned the Position Update, TCP Push:

Email and TCP Pull: HTTP application models. This profile has so many applications because it is typically only the tactical commanders who perform these kinds of events frequently and simultaneously.

b. Fire Support

This application profile models the types of traffic introduced to the network by a forward observer for fire support. It has the exact the same requirements as the Tactical Commander profile, but because they are two very different jobs, a separate profile was created, in order to allow scenario adjustments to the behavior of one type of node without affecting the behavior of the other.

c. JTAC

This application profile models the types of traffic introduced to the network by a JTAC (a special type of forward observer). For our simulations it will only be assigned the video application model; however, in some scenarios the JTAC will have similar requirements to those of the Fire Support profile.

d. Gateway

For our simulations, the network gateway node will not actually run any applications itself, so its application profile will not actually be assigned any application models. This node will, however, be the source or destination of traffic generated by many of the non-gateway nodes, such as the Unicast to Gateway, TCP Push: Email, and TCP Pull: HTTP application profiles.

e. Position Update

This application profile will mimic the type of traffic introduced to the network by a node reporting its position to the gateway node. Each network node will be assigned to this profile, and it will produce traffic concurrently with other applications running at the same node. It was kept separate from other profiles in order to create baseline simulations that have position reporting as the only traffic introduced across the network.

f. Short Message

This profile will represent nodes that are sending simple text messages across the network. It was kept separate from other profile, in order to create additional baseline simulations that have small text messages as the only traffic introduced across the network.

Now that we have discussed how our network simulation software works, its capabilities and limitations, and how actual communication devices and traffic loads can be translated into software models, we will now begin our analysis of each radio network's performance under different simulated network conditions.

The following chapter will discuss the network performance data collected from software simulations involving both SINCGARS and EPLRS networks, attempt to explain each network's performance and compare the results to similar simulations collected in [1], involving CDR networks.

V. EXPERIMENTAL SETUP AND RESULTS

A. EXPERIMENTAL SETUP

In this chapter, we will compare the network simulation results of scenarios involving SINCGARS and EPLRS tactical radios to those involving the CDR, as presented in [1]. Once the results are presented, we will discuss the varying performances of each tactical radio model in networks of similar size and under similar traffic loads. All of the tactical radio node models used in these scenarios are unaltered and reflect the general performance characteristics of actual tactical radios.

For each radio, we will run simulations configured for both a platoon-sized network (6 nodes) and a company-sized network (20 nodes). Each network will be populated with five levels of gradually increasing traffic loads, each of which reflect the estimated traffic loads of various combinations of tactical data applications. The applications included in each of the five load levels are shown in Table 3, with the load level descriptors listed in the leftmost column and the available application profiles listed in the top row. The application profiles running at each load level are indicated with marks placed along each load level's row, under each application profile included at that level.

	Tactical Commander	Fire Support	JTAC	Position Update	Short Message
Position Update Only				X	
Short Message Only					X
Commanders and Position	X			X	X
Currently Deployed Applications			X	X	X
All	X	X	X	X	X

Table 3. Application Profile Sets For Network Simulations (From [1]).

For the simulations involving SINCGARS networks, each node consists of a SINCGARS radio terminal model that provides the physical layer component of each node. This SINCGARS radio terminal connects to generic Ethernet workstation model, via a SINCGARS Internet Controller model, which provides an appropriate interface between the two devices. All network traffic is generated and received by the generic Ethernet workstation at each node, with the exception of the Gateway node, where a generic Ethernet server model is used instead. Each SINCGARS radio terminal is set to the maximum 16 Kbps throughput and all wired connections between models is 100BaseT or higher. All SINCGARS nodes access the transmission medium as soon as they have messages to transmit, essentially implementing a regular ALOHA MAC protocol across the network, and the only type of quality control implemented across this network is the use of TCP by select applications. Otherwise, if a message's initial transmission experiences a collision, it is never retransmitted by the node.

In each EPLRS network simulation, each node consists of an EPLRS radio terminal model that provides the physical layer component for each node. It connects to a generic Ethernet workstation model through an EPLRS router model interface. All network traffic is generated and received by the Ethernet workstations, with the exception of the Gateway node, where a generic Ethernet server model is used instead. All nodes are members of a CSMA needline assigned the maximum of 4 usable LTSs, and is set to use waveform 4, which allows a maximum throughput of 311 Kbps (the waveform with the highest throughput of all the available JCSS EPLRS model waveforms). Each node is set to allow a maximum of 6 hops, which, given the nature of the EPLRS CSMA needline implementation, reduces the effective throughput utilization to 1/6 of the maximum needline throughput, or 51.83 Kbps for our scenario.

Unlike the SINCGARS, each EPLRS node will first check to see if there are any other messages being transmitted before beginning to send their own message, implementing a CSMA/CA MAC protocol. If the node senses another transmission, it will queue its message and check the medium again later (after a randomly generated delay). Each time it senses that the medium is busy, it will choose an incrementally longer back-off delay, until it senses no other transmissions, at which time it will start

sending its own message. Since we could not find a maximum needline hold time setting for the JCSS EPLRS node model, we will assume that each node in each of our simulations controls the transmission medium (potentially indefinitely) until it is done transmitting.

For both sets of simulations, each node will remain stationary, and each node will be within transmission rage of every other node. While the specific scenario being simulated in this thesis does not require the transmission of more than one hop, it assumes that the nodes that support multi-hop capabilities are configured to allow the maximum allowed hops, in preparation for follow-on mobile operations, where nodes may not be within one hop of each other. Since the EPLRS nodes do not dynamically adjust the hop count settings based on perceived network topology, it is fair to assume that even when each node is within one hop from all other nodes, its performance will still be limited by the network's maximum hop count settings. Also, different needline configurations handle multi-hop delivery differently, and may produce different performance ratings. For comparison purposes, we attempted to choose the type of needline that most closely resembles the configuration settings of the CDR analyzed in [1], in order to produce results that demonstrate the impact of the general differences between each device. The performance of different needlines and more dynamic network scenarios, where the relative distances between each node is constantly increasing and decreasing are left for future research.

The following are explanations of each traffic load level:

1. Position Update Only

This application traffic load level consists of messages that are regularly transmitted once every minute, and represent position location messages being sent by all nodes attached to the network to the network's gateway node. This scenario is included to verify that correct connectivity configurations exist between each node in the simulation. It also provides a baseline for performance comparisons to other radios operating under the same load level and to simulations of the same network operating under more demanding load levels.

2. Short Message Only

This application traffic load level consists of slightly larger text messages that are transmitted at exponentially varying times across the network. This application represents text messages being sent between two nodes attached to the network. This load level is included to show each networks ability to handle simple text messaging traffic that is currently in use by existing tactical radio networks.

3. Commanders and Position

This application traffic load level includes all of the modeled tactical data applications except for streaming video. This level is included to demonstrate each network's ability to function when supporting all but the most bandwidth intensive application.

4. Currently Deployed Applications

This traffic load level is meant to model each radio's ability to handle all of the tactical data applications currently in use by tactical mobile nodes. Since many of these applications are accessed at a single node through separate networked radio devices, if a network composed of a single type of radio device is shown to be able to support all of these applications at once, then it would be reasonable to think that it would be possible to streamline each node's equipment load to only one radio device.

5. All

This traffic load level includes some applications that are not currently available to users operating at the lowest levels of tactical command (i.e., E-mail and IRC), and it serves as an ultimate comparison scenario for a particular radio network's ability to support an ideal application environment for users at the platoon and company levels.

B. PLATOON SIMULATIONS

Our platoon-sized network simulations consist of six radios: one platoon commander, three squad leaders, one mortar platoon, and one gateway node. Figure 9 depicts the location of each radio model within the simulated network.

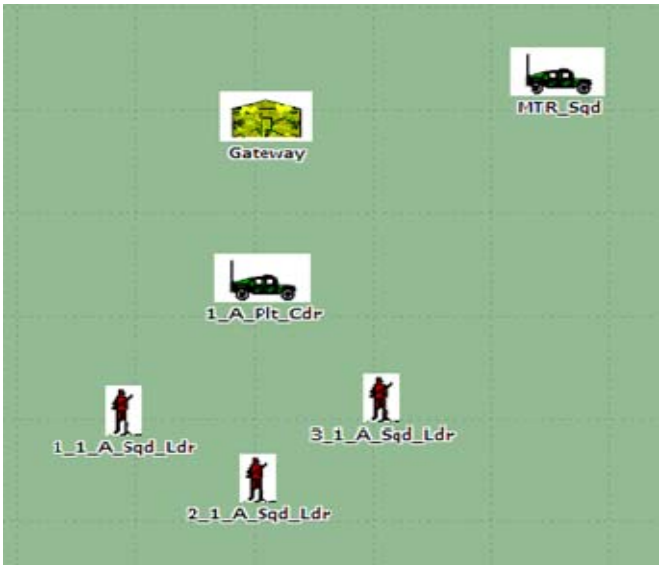

Figure 9. Platoon Network Layout

Each node attached to the network is limited to transmitting the types of traffic allowed by the application profile associated with that particular node during each scenario. So, depending on the scenario some nodes may introduce different types and quantities of network traffic than the other nodes. The applications associated with each node attempt to mimic the types of traffic that would be reasonably expected from the job assigned to each node. For example, a squad leader is only allowed to use the position update and short message applications, the mortar platoon can only use the position update, short message and fire support applications and the platoon commander can use all application profiles. The gateway node does not initiate the use of any applications. It only acts as a source for streaming video, a recipient of position report data and source of TCP application traffic, for nodes sending TCP traffic that is routed through the gateway.

1. Position Update Only

All three networks were able to successfully support this network simulation scenario. The SINCGARS and EPLRS networks each achieved a 100% message delivery success rate, and the results presented in [1] show that the CDR network achieved a 99.6% message delivery success rate, missing only four messages during the 15 hours of total simulation time.

The fact that the CDR did not achieve a 100% success rate is a little surprising. Since all application traffic being introduced into the Position Update Only scenario enters the network at a constant rate (each node transmitting the same message in a 60 second cycle), and all nodes are stationary, if the collisions were caused by two nodes attempting to introduce new traffic at the same time, we would expect to see these interferences occurring much more regularly (once per 60 second cycle, or 180 collisions per 3 hour simulation), producing more than a total of only four missed messages.

Additionally, if the collisions were caused by interfering message relay transmissions from multiple CDR nodes, we would expect this type of interference to occur at a constant rate, as well. Since there were no other generic sources of environmental noise introduced into this scenario, the only source of signal noise that can affect each simulated transmission are transmissions from other nodes. With such a high number of different "simultaneous" CDR relay events being processed in serial by the discrete event kernel, it is possible, but unlikely, that these transmission collisions could be attributed to errors introduced by the simulation kernel's processing order of CDR message relay events.

The most likely source of these collisions is caused by the transmission pipeline's placement of simulated bit errors within each packet. These bit errors are partially determined by the software's pseudorandom number generator, so if there were slightly overlapping transmission simulation events that occurred regularly throughout the scenario, then it would be possible for the network to only experienced a small number of unrecoverable bit errors, due to the specific placement of each bit error within certain simulated transmissions.

Nonetheless, it is not surprising that both the SINCGARS and the EPLRS networks were able to support the application being run during this scenario. The results of both networks area presented in greater detail in the following two sections.

a. SINCGARS Performance

Our SINCGARS network yielded a 100% success rate in message delivery by all nodes transmitting across the network. Of the 4475 total traffic message events that were created during the 15 hours of simulation time, every message was delivered successfully. This performance was expected, because all radios were transmitting at the same constant rate and each radio started their transmissions at staggered simulation times, ensuring that no two radios would be transmitting simultaneously.

During this scenario, neither the transmission nor the reception rates at each SINCGARS node ever went above 624 bits per second. As shown in Figures 10 and 11, they both have an average throughput of around 300 bps, which is well below the 16 Kbps maximum data rate of the radio. These figures will be used for comparison to those generated by other simulation runs discussed later in this chapter.

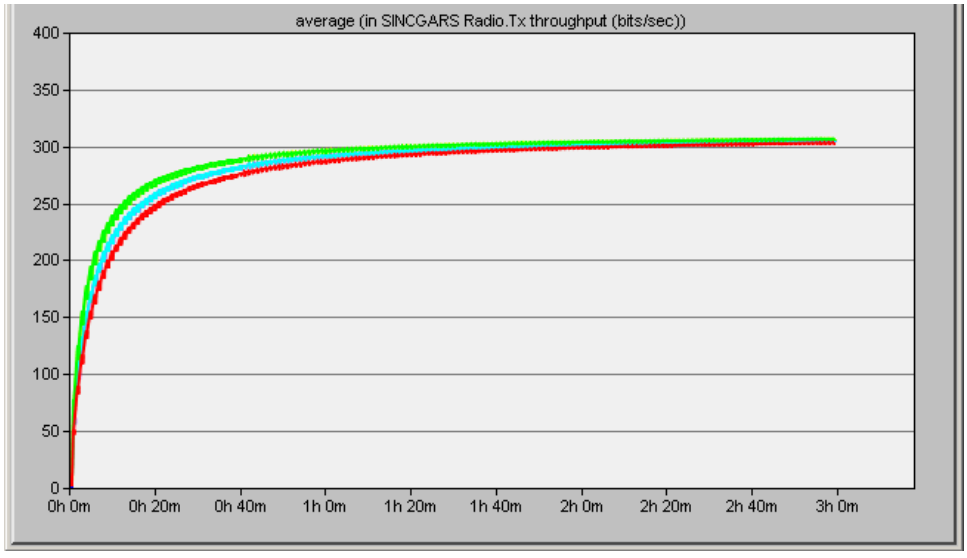

Figure 10. SINCGARS Platoon (Position Only): Average Tx Throughput

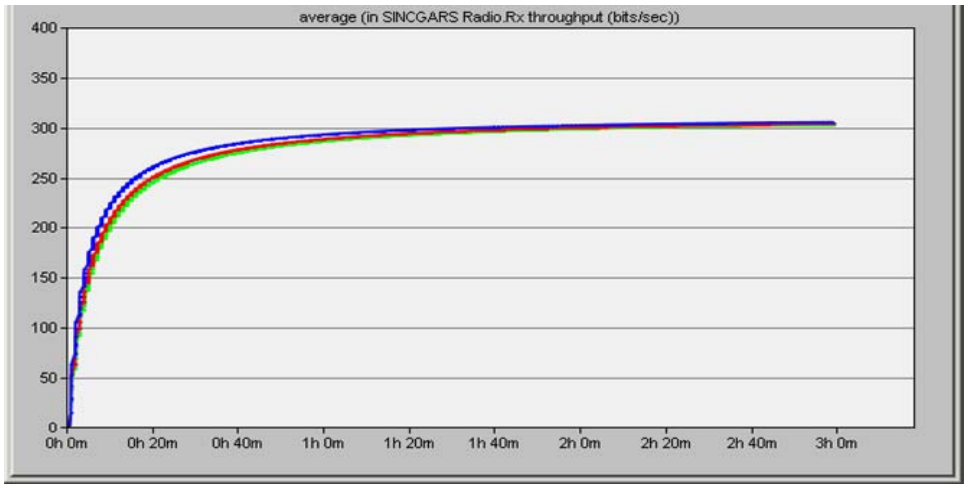

Figure 11. SINCGARS Platoon (Position Only): Average Rx Throughput

b. EPLRS Performance

Our EPLRS network also yielded a 100% message delivery success rate during this simulation scenario. Of the 4475 total traffic message events that were create during the 15 hours of simulation time, every message was delivered successfully. Again, this is not entirely unexpected, because each radio transmitted their messages at the same constant rate, ensuring that no two radios would be transmitting simultaneously.

The transmission and reception rates at each EPLRS node never go above 250 and 355 bps, respectively. Figures 12 and 13 show that the average transmit throughput of four of the nodes was roughly 45 bps and the average receive throughput of the same four nodes was between 43 and 45 bps at. Both the Gateway node and MTR Squad node average throughputs vary noticeably from the other four nodes.

Since the Gateway did not originate any application traffic, it rarely blocked the transmissions from other nodes by attempting to transmit itself, so its receive throughput measured higher than all of the other nodes in the network (the top line in Figure 13). We do notice, however, that the Gateway does transmit a small amount of traffic (bottom line in Figure 12). This is simply the EPLRS device communicating routing information to its neighboring devices, and demonstrates the small amount of network overhead required for the maintenance of even simple EPLRS networks.

The MTR node transmits much less data than the other non-Gateway nodes on the network (second line from the bottom in Figure 12). This is because this node is further away from the other nodes than the Gateway node. Since the Gateway node is the destination for all application traffic generated during this scenario, the MTR node knows that it is not as close to the other nodes as the gateway, so it chooses not to relay any of the other nodes' transmissions. Since the four other nodes are roughly the same distance away from the Gateway, they most likely attempt to relay most of each other's transmissions to the Gateway, which explains their almost identical average transmission throughputs (lines overlapping at the top of Figure 12).

Regardless of the differences between each of the nodes, all of these rates are well below the 51.83 Kbps maximum achievable data rate of our CSMA needline. These figures will be used for comparison to those generated by other simulation runs discussed later in this chapter.

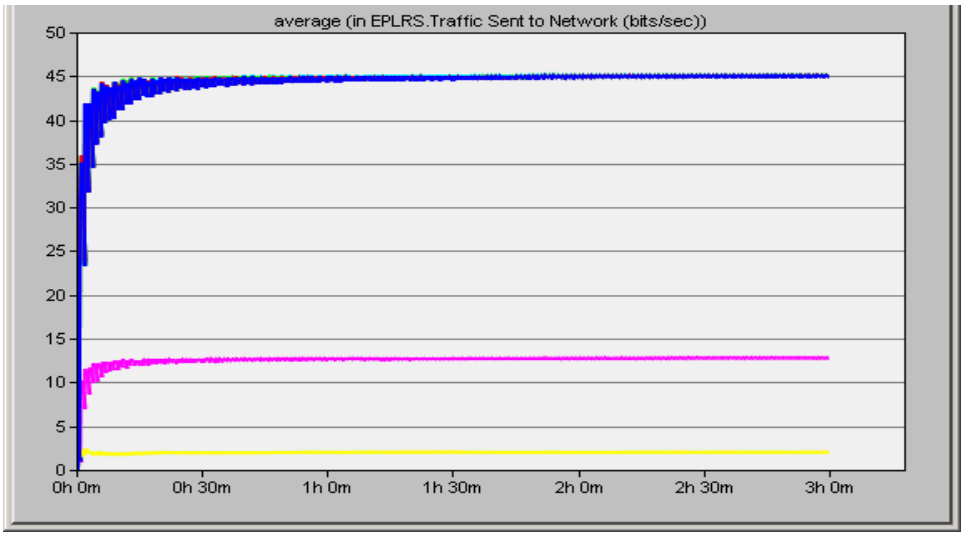

Figure 12. EPLRS Platoon (Position Only): Average Tx Throughput

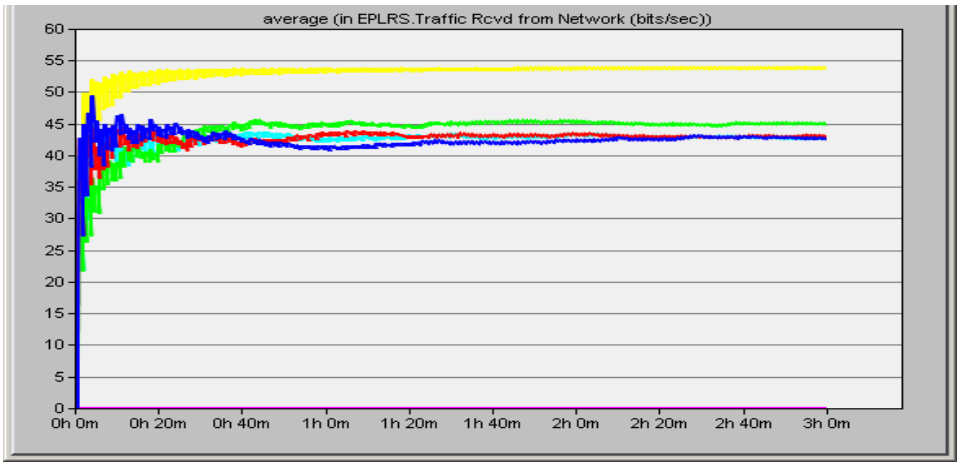

Figure 13. EPLRS Platoon (Position Only): Average Rx Throughput

Notice that the SINCGARS network averaged an overall significantly higher transmit and receive throughputs than the EPLRS network. These differences are due to the higher throughput of the EPLRS device. Since the transmission rate of the EPLRS is much higher than that of the SINCGARS, it spends more time sitting idle than the SINCGARS radio, which means that under the same traffic conditions, over any period of time the longer idle times will bring down the throughput utilization averages. This does not mean that the nodes on the EPLRS network transmitted fewer or smaller application messages.

2. Short Message Only

Only the CDR and SINCGARS networks were able to successfully support this simulation scenario. The CDR and SINCGARS networks achieved 99.6% and 99.9% delivery success rates, respectively, and the EPLRS network achieved a 60.2% delivery success rate, demonstrating the EPLRS surprising inability to support this scenario using the CSMA needline.

In the CDR network, the results presented in [1] show that no more than two retransmissions were required across the 15 hour simulation, with only 3 message transmissions taking longer than about a second (but no more than 3 seconds). It required between one and two retransmissions per hour and saw an average TCP latency of 0.408 seconds.

The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

a. SINCGARS Performance

Our SINCGARS network yielded a 99.9% message delivery success rate by all nodes attached this network. Of the 1756 traffic total message events that were created, only two messages were not delivered successfully. Even though this scenario produced less traffic than the Position Update Only scenario, because each node was generating messages of greater size (160 Bytes versus 50 Bytes) at random intervals, instead of smaller messages at constant, offset intervals, it is not surprising to encounter a small number of collisions caused by simultaneously generated transmissions.

Despite the larger message size and the greater amounts of overhead network traffic caused by the use of TCP messages for this application's transmissions, Figures 14 and 15 show that the average transmit and the receive throughputs actually show minor decreases from those seen in the previous scenario. This is likely a result of the lower quantity of messages transmitted across the network in this scenario, causing the increased amount of idle time for each radio to lower the overall average throughputs.

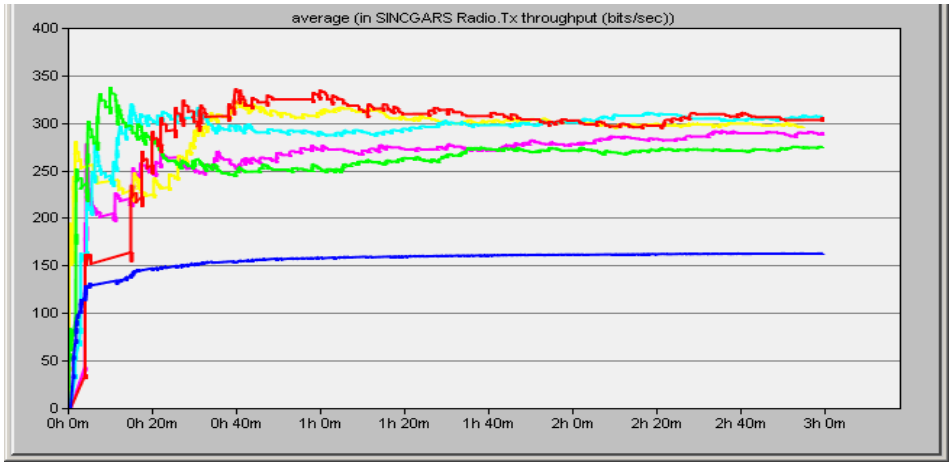

Figure 14. SINCGARS Platoon (Short Message Only): Average Tx Throughput

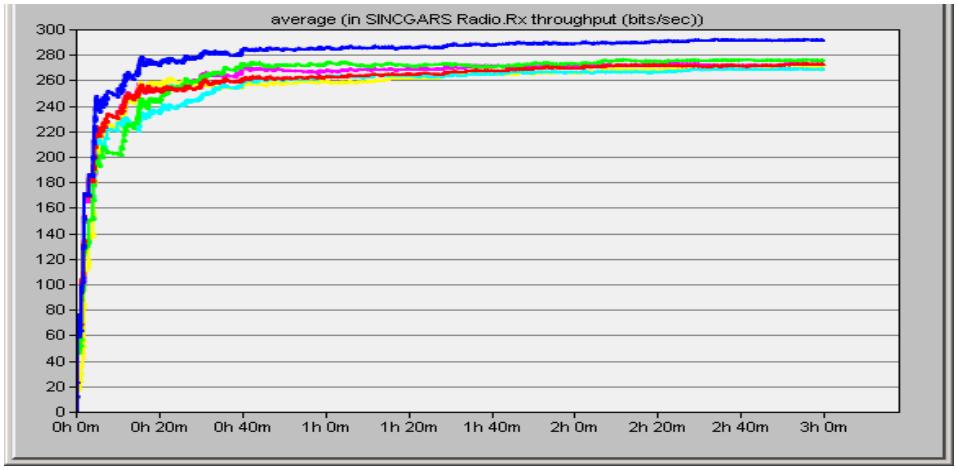

Figure 15. SINCGARS Platoon (Short Message Only): Average Rx Throughput

As shown in Figure 16 and 17, the average TCP delay was 0.772 seconds (almost twice as long as the CDR TCP latency of 0.408 seconds) and, there were approximately 6 TCP retransmissions per hour (three times that of the CDR), with only two transmissions that needed to be retransmitted twice.

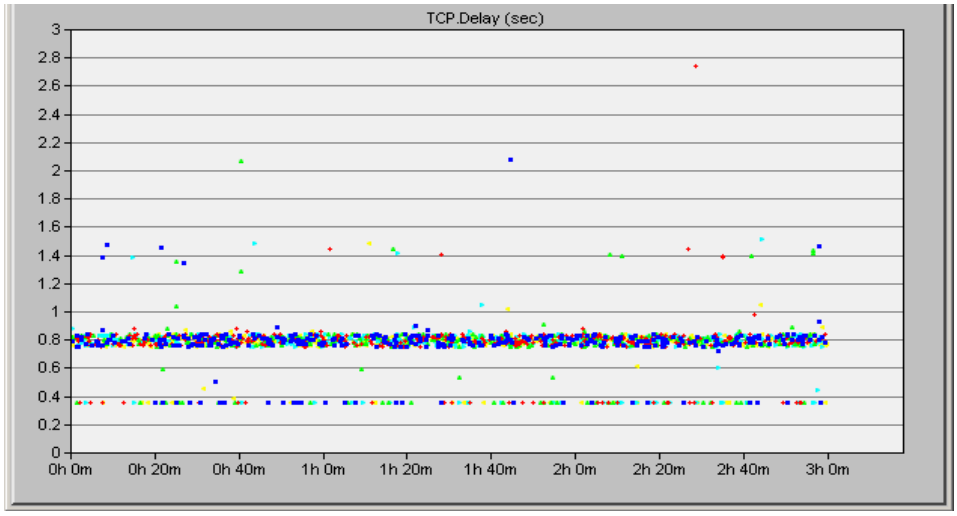

Figure 16. SINCGARS Platoon (Short Message Only): TCP Delay

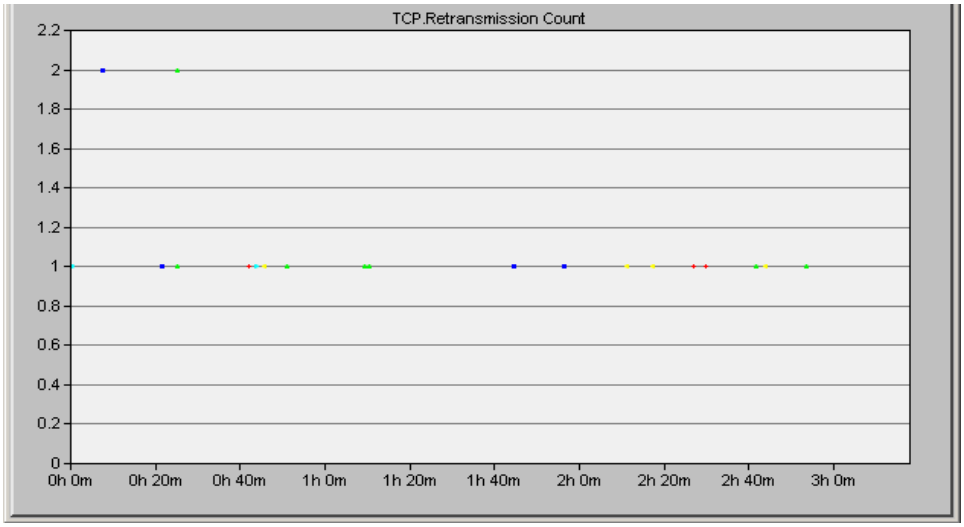

Figure 17. SINCGARS Platoon (Short Message Only): TCP Retransmission Count

b. EPLRS Performance

Our EPLRS network had an overall message delivery success rate of 60.2% during this simulation scenario. Of the 1653 total traffic message events that were created, 658 messages were not delivered successfully. This was significantly less successful than both the SINCGARS and CDR networks, and would not be acceptable performance metrics for use by actual military units.

As depicted in Figures 18 and 19, we see that the average transmission rates for each node range between 120 and 140 bps, and the average reception rates range between 170 and 225 bps. This is roughly four times greater than that of the Position Update Only scenario, but still well below the 51.83 Kbps maximum data rate of our CSMA needline. Again, we see a noticeable difference between the amount of traffic sent and received by the MTR Squad (bottom line in Figure 19 and 20) and the rest of the nodes. This is caused by the same reasons as those mentioned in the Position Update Only scenario discussion.

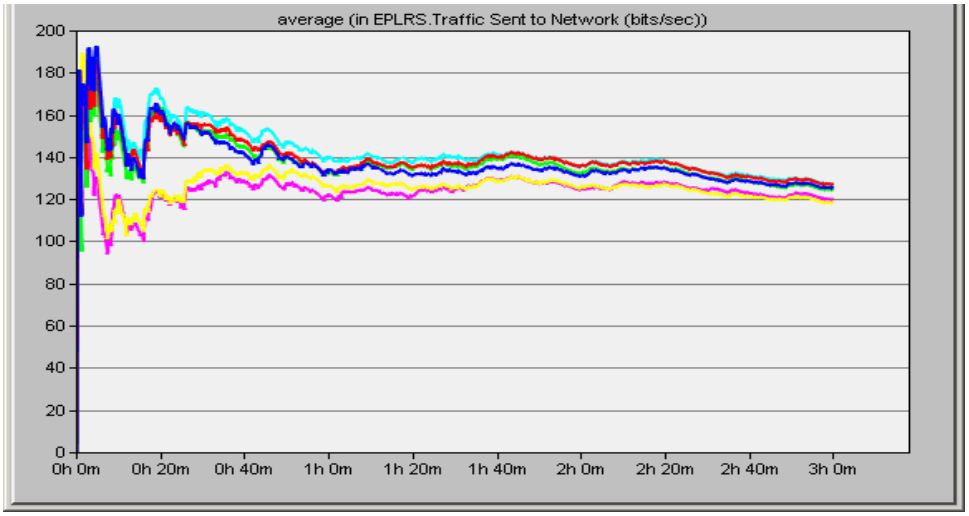

Figure 18. EPLRS Platoon (Short Message Only): Average Tx Throughput

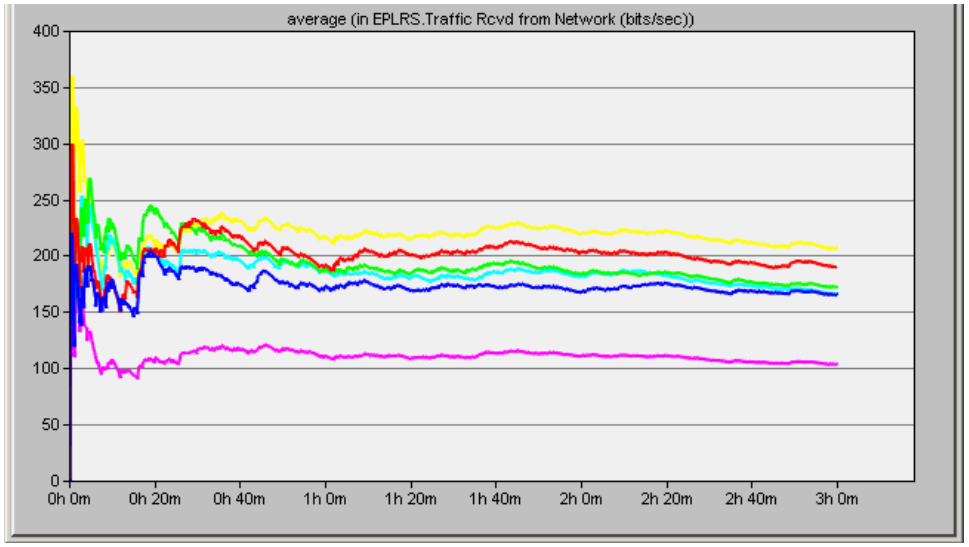

Figure 19. EPLRS Platoon (Short Message Only): Average Rx Throughput

As shown in Figure 20, the average TCP delay had a maximum delay of 76.813 seconds, and averaged 1.1472 seconds, which is almost three times greater than the 0.408 second average delay for the CDR network.

Figure 21 shows that there were significantly more TCP retransmissions than in the SINCGARS. The average retransmission count was 2.75 (SINCGARS averaged 1.25), with a peak count of 12 attempts. The increased number of retransmissions is most likely due to there being a high number of instances where

multiple nodes were attempting to introduce traffic at the same time. Remember that the EPLRS CSMA needline multi-hop implementation forces each node to delay their transmissions until the previously transmitted signal has had the chance to travel the maximum number of hops. Since each node attached to the CSMA needline is forced to wait until the same time after each transmission and transmission relay before introducing new traffic (for a 6 hop network the transmission originator waits 5 time slots, the first relay node waits 4 time slots, etc.), the chances of multiple nodes attempting to introduce traffic immediately at the end of this CSMA needline imposed wait period are fairly good. Since there was no forced wait time for the SINCGARS node transmissions, this may account the fewer number of transmission collisions.

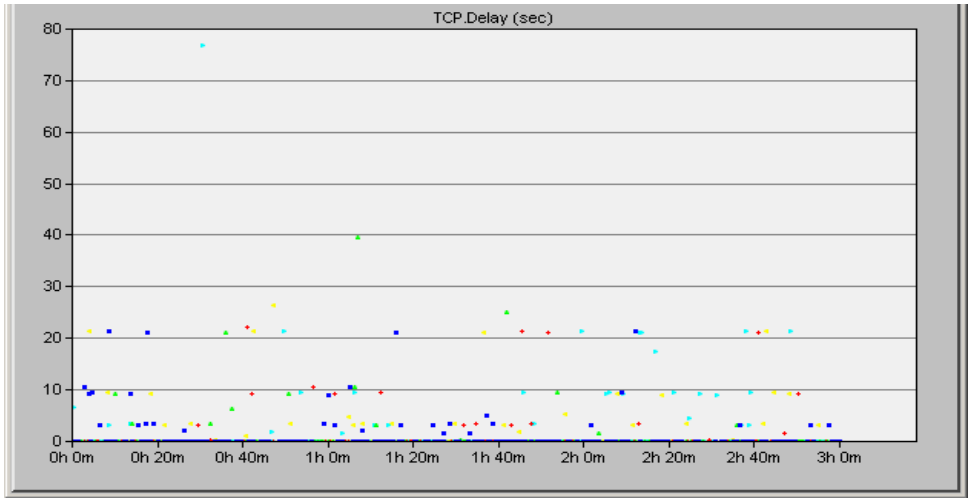

Figure 20. EPLRS Platoon (Short Message Only): TCP Delay

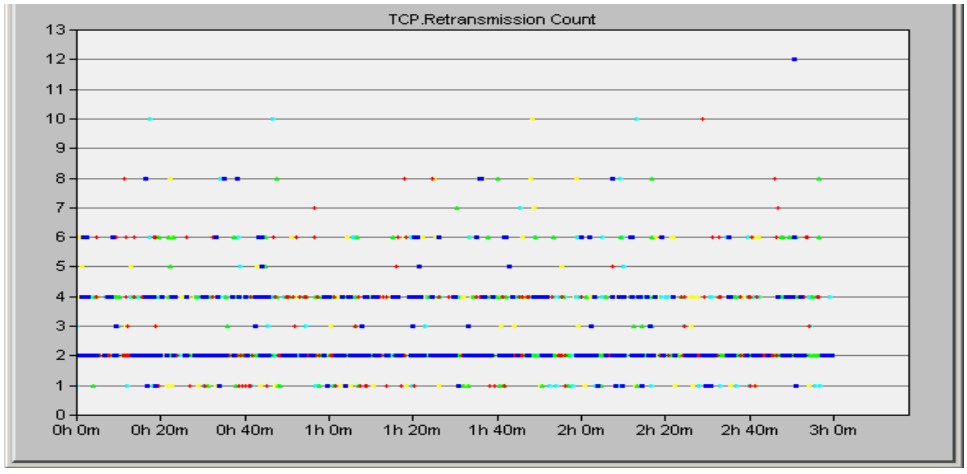

Figure 21. EPLRS Platoon (Short Message Only): TCP Retransmission Count

In Figure 22, we show data from three separate EPLRS nodes representing the amount of traffic each node is introducing into the network (the top three lines) and the amount of traffic being generated by applications running at each node (the bottom three lines). For each of these nodes, the amount of traffic being transmitted across the network is roughly four times as great as the traffic being generated, illustrating the increased traffic load created by using TCP, relaying other nodes' transmissions, and the small amount of overhead involved in each node maintaining its routing information for the network.

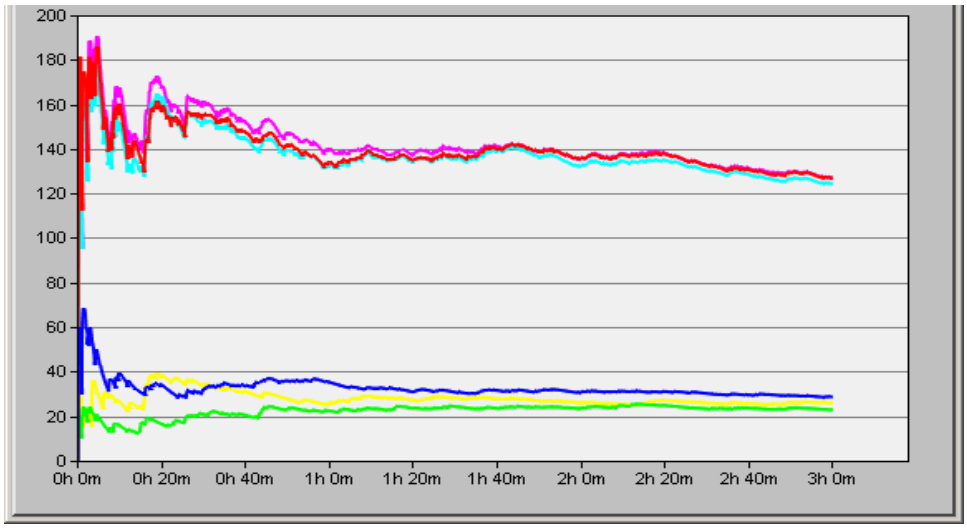

Figure 22. EPLRS Platoon (Short Message Only): Relay vs. Non-Relay Traffic

We ran this simulation again using UDP, instead of TCP, and of the 1745 total UDP messages, all but two messages were delivered successfully (a 99.89% success rate). Obviously, the additional overhead incurred by the use of TCP had a significant impact on the effectiveness of this network.

Aside from switching our application's messages from TCP to UDP, it might be also possible to use different needline configurations of the EPLRS device to achieve acceptable results for similar traffic loads across an EPLRS network (i.e., reducing hop count or altering waveform type), but that will be left to future work.

3. Commanders and Position

A summary of each network's performance under this scenario is shown in Table 4. Surprisingly, only the SINCGARS network was able to fully support all five node applications, despite it having the lowest overall transmission rate of the three network devices. The results presented in [1] show that the CDR network easily supported three of the applications, but failed to provide adequate HTTP and Email support. The EPLRS provided better HTTP and Email support than the CDR, but at the expense of the other three applications. The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

%	CDR	SINCGARS	EPLRS
Position Update	99.5	100	94.49
Short Message	100	98.13	66.73
IRC	100	100	86.95
HTTP	39	100	55.83
Email	39	100	55.83

Table 4. Commanders and Position Application Success Rates (Platoon).

a. SINCGARS Performance

Our SINCGARS network running the Commanders and Position scenario achieved the highest success rate in message delivery by all nodes transmitting across the network. Of the 17412 total traffic message events that were created, only 19 messages were not delivered successfully. However, it is worth mentioning that this simulation did not include the introduction of normal tactical voice traffic, which would have surely degraded these performance results significantly, since the radio does not effectively support simultaneous transmission of both data and voice across a single channel.

Under this scenario's network load, we see that both the transmission and reception rates increase, as expected with the introduction of additional network traffic. Since all of the additional traffic is between the Gateway and Platoon Commander nodes, we notice that the transmission rates of the Squad Leader nodes do not increase, but the

amount of traffic introduced onto the network by the Gateway (top line) increases to almost 5 times that of the other nodes. Additionally, the amount of traffic received by all of the other nodes triples, as displayed in Figures 23 and 24.

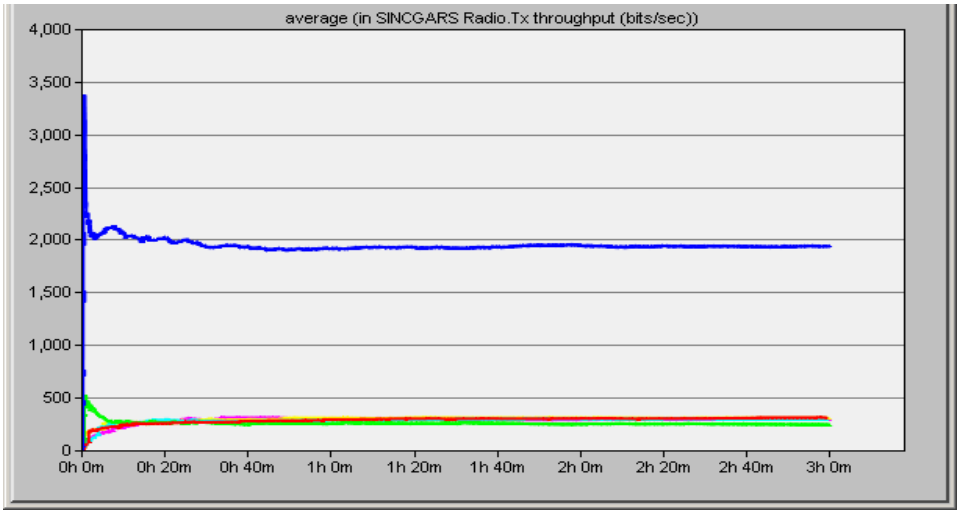

Figure 23. SINCGARS Platoon (Commanders & Position): Average Tx Throughput

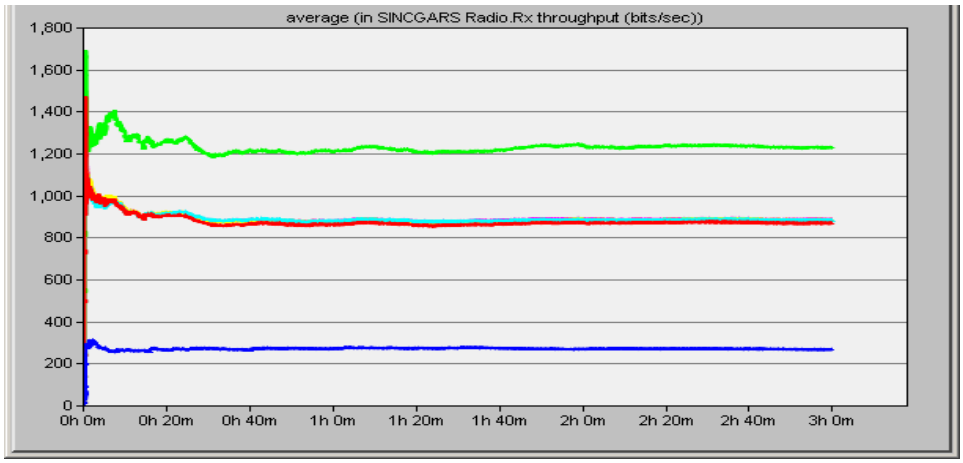

Figure 24. SINCGARS Platoon (Commanders & Position): Average Rx Throughput

Overall, the TCP statistics reflect the impact of increased network traffic, showing increases in delays and retransmission counts. Figure 25 and 26, show that the average TCP delay was 1.12 seconds, almost double what was experienced in the Short Message Only scenario, and the average TCP retransmission count was 1.27, only slightly higher than the previous scenario, with none being retransmitted more than 4 times (double the previous maximum retransmission count of 2 times).

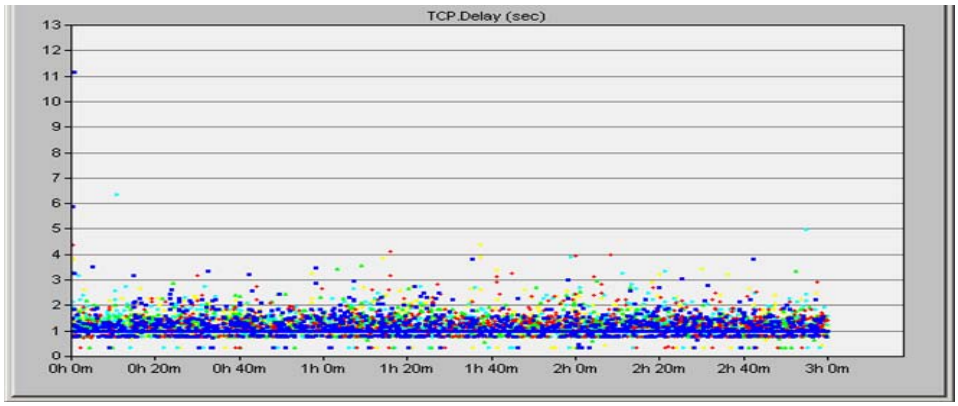

Figure 25. SINCGARS Platoon (Commanders & Position): TCP Delay

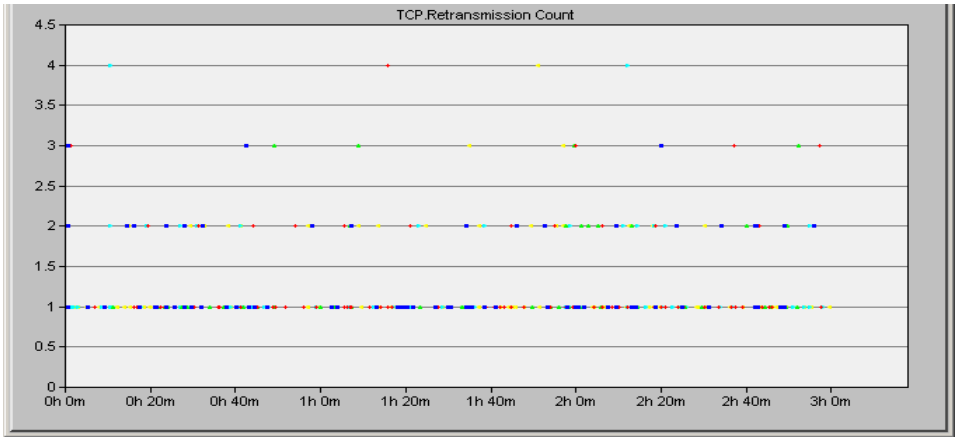

Figure 26. SINCGARS Platoon (Commanders & Position): TCP Retransmission Count

b. EPLRS Performance

While the EPLRS network in this scenario showed slightly improved Short Message application success rate (more than 6 percent higher), it still did not come close to being able to adequately support four of the five applications, the exception being the Position Update application. The reason we saw this change in the performance of the Short Message application was due to different pseudorandom numbers controlling the actual transmission timing of each of these messages. Even though each simulation run was started with the same seed as those from the previous scenarios, since there are more applications using the same pseudorandom number generator to determine their transmission times, the transmissions that occur later in the simulation will be triggered by different numbers than they were in the previous scenario.

This is one quirk of using software-based simulations, which is discussed in greater detail in [15] and [16], and is not intended to imply that the throughput of one application should increase with greater overall resource demand.

Under this scenario's network load, we see that both the transmission and reception rates increase, as expected with the introduction of additional network traffic. Since all of the additional traffic is between the Gateway and Platoon Commander nodes, we notice that the transmission rates of the Squad Leader only double, but the amount of traffic introduced onto the network by the Gateway increases to almost four times that of the other nodes. Additionally, the amount of traffic received by the Squad Leaders double and the amount received by the Gateway and MTR Squad increase by a factor of 10, as shown in Figures 27 and 28.

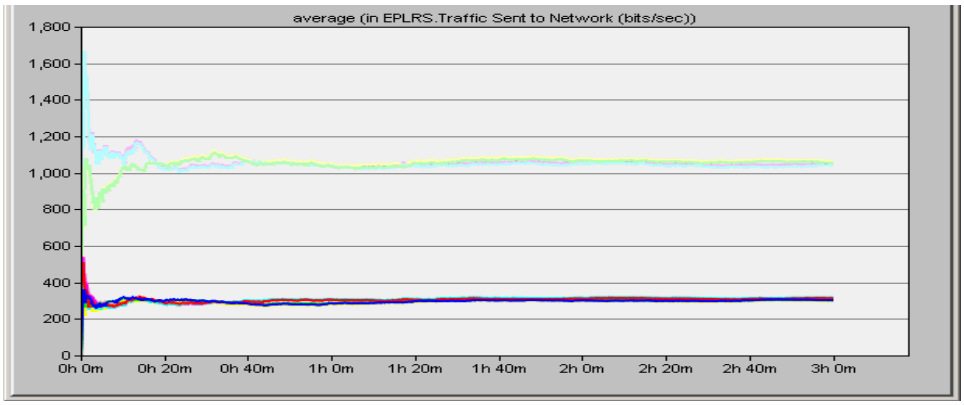

Figure 27. EPLRS Platoon (Commanders & Position): Average Tx Throughput

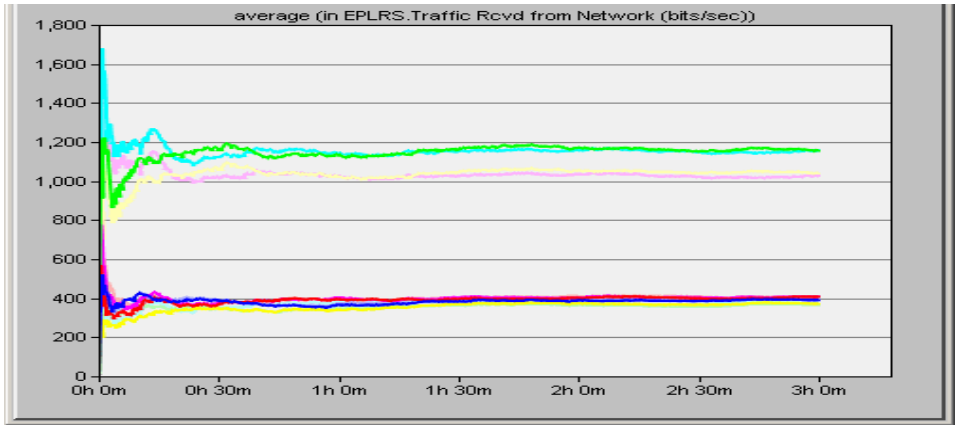

Figure 28. EPLRS Platoon (Commanders & Position): Average Rx Throughput

Overall, the TCP statistics reflected the impact of increased network traffic, showing increases in delays and retransmission counts. As shown in Figure 29, the average TCP delay was 0.321 seconds for all but one of the simulation runs for this scenario. Figure 30 shows that one of our simulation runs achieved TCP delays of over 6000 seconds. That is a delay of over 100 minutes, which would be entirely too long for any of the applications running across our network.

The long delay seen in this simulation run was probably caused by one node's transmit queue becoming backed-up to a level that causes significant delays. Since we are assuming that there is no maximum holding time limit for a single node to utilize the CSMA needline, once a node has the needline, it will not let it go until it has emptied its transmit queue. This means that there is a chance of a single node having to wait indefinitely for access to the needline resources. This circumstance is mitigated in real life by the EPLRS "maximum hold time" setting, which sets a ceiling for the amount of time one single node can maintain control over a single needline's network resources. Because this setting was not available for our JCSS EPLRS model, we see one potential result of having the maximum hold time set to infinity.

Figure 31 shows that the average TCP retransmission rate was 2.24 (almost twice that of the SINCGARS network), with a majority of transmissions being retransmitted between 1 and 6 times. The maximum retransmission count of 15 retransmission attempts was almost four times greater than the maximum count from the SINCGARS network.

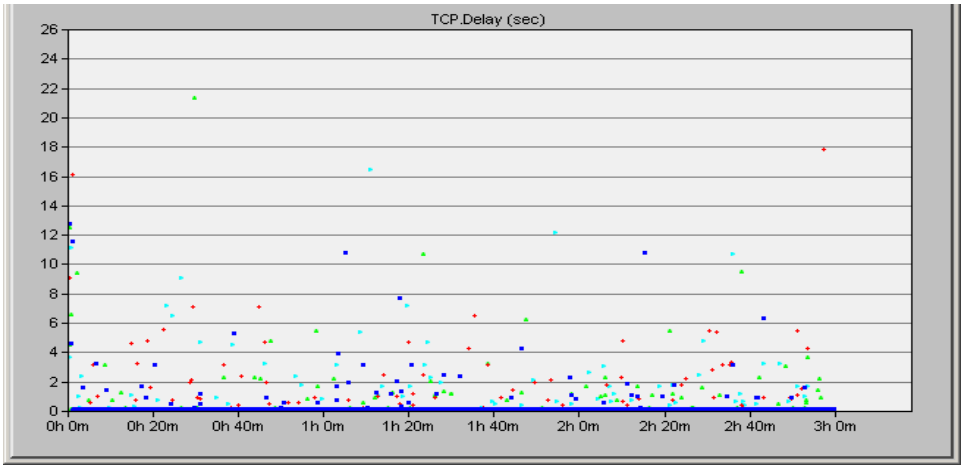

Figure 29. EPLRS Platoon (Commanders & Position): TCP Delay

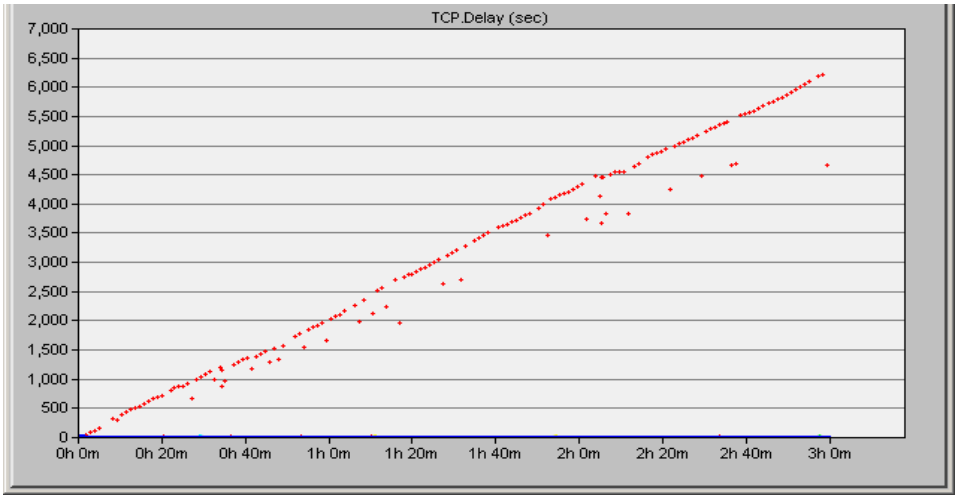

Figure 30. EPLRS Platoon (Commanders & Position): TCP Delay

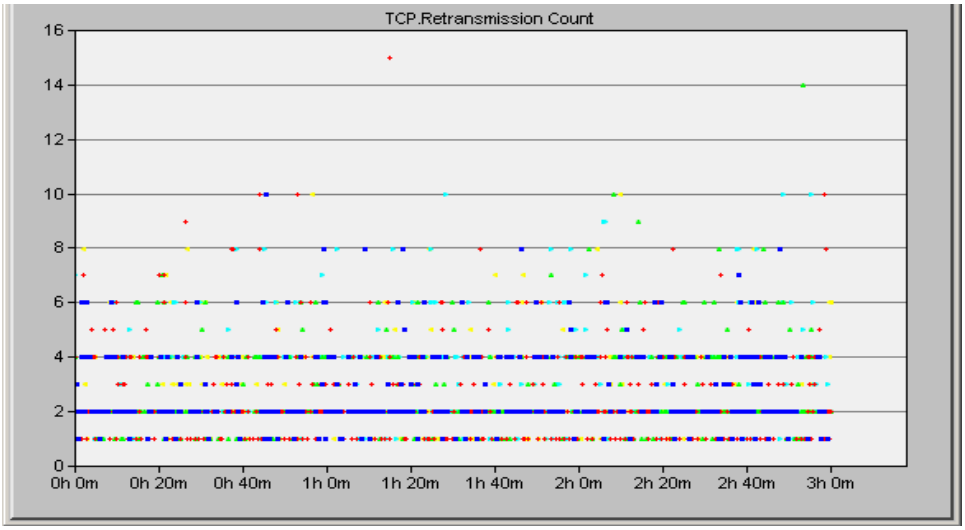

Figure 31. EPLRS Platoon (Commanders & Position): TCP Retransmission Count

4. Currently Deployed Applications

A summary of each network's performance under this scenario is shown in Table 5. Using the results presented in [1], we see that only the CDR network was able to fully support all three applications. This is most likely because of its use of cooperative diversity, which allows each node to interpret multiple instances of the same transmission as a single transmission, that is, to mitigate collisions by treating identical receipts as 'non-colliding transmissions. Since both the Position Update and Video applications produce a high volume of regularly transmitted messages, the likelihood of these

transmissions (and their respective relays) colliding with each other is high. The ability of the CDR to overcome these types of transmission collisions and successfully allow these messages to be delivered is evident when considering the simulation results.

Overall, the SINCGARS network was marginally able to support the applications run in this scenario, and the EPLRS network failed to support all but the Video application. The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

%	CDR	SINCGARS	EPLRS
Position Update	99.6	100	24.24
Short Message	99.6	72.11	60.58
Video	99.9	80.7	98.66

Table 5. Currently Deployed Applications Application Success Rates (Platoon).

a. SINCGARS Performance

Our SINCGARS network was only able to support the Position Update application during this simulation scenario. However, it is a little surprising that none of the Position Update messages collided with the Video application transmissions. Since both applications are generating constantly spaced transmissions, Video once every 0.5 seconds and Position Update once every minute for each node, it would seem likely that at least one Position Update message would run into one of the 21600 video messages, especially since both applications are set to begin their transmissions at the very beginning of a simulation-time second. The significant decrease in support of the Short Message application is likely due to increased traffic collisions with the constantly streaming Video application.

Under this scenario's network load, Figures 32 and 33 show that the average transmission throughput for each radio remains roughly the same, where the reception throughput almost doubles. Since the only the gateway node has an increased

transmission burden of sending the Video application messages, it makes sense that the reception throughputs of the other nodes increases, since all other nodes will be receiving more data when sitting idle.

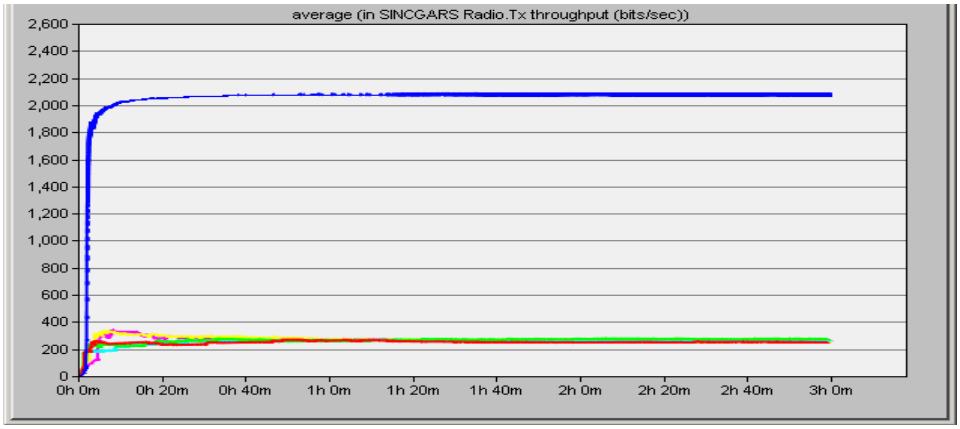

Figure 32. SINCGARS Platoon (Current Applications): Average Tx Throughput

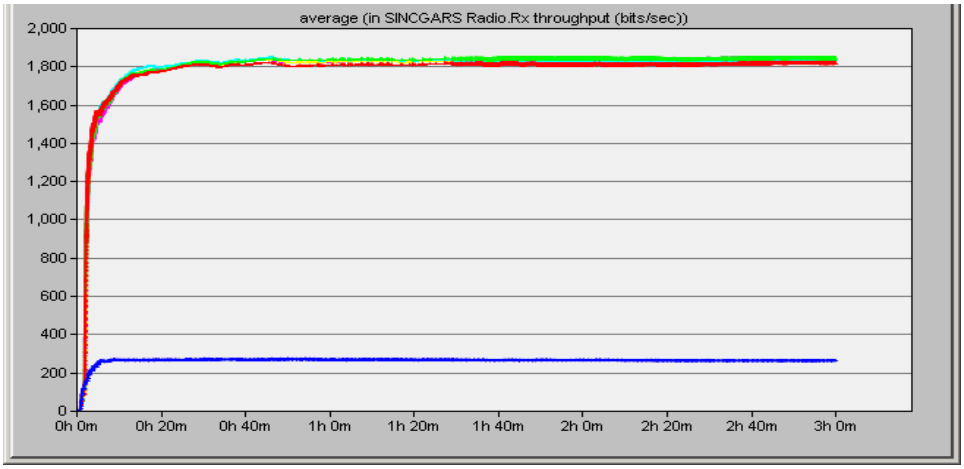

Figure 33. SINCGARS Platoon (Current Applications): Average Rx Throughput

Overall, the TCP statistics reflected the impact of the increased network traffic, showing increases in delays and retransmission counts. As shown in Figure 34 and 35, the average TCP delay was 1.96 seconds—almost double what was experienced in the previous scenario by this network, but the maximum TCP delay decreased from 11.15 seconds to 8.51 seconds. The average TCP retransmission count was 1.29, only marginally higher than the previous scenario, with a maximum TCP retransmission count of six retransmissions.

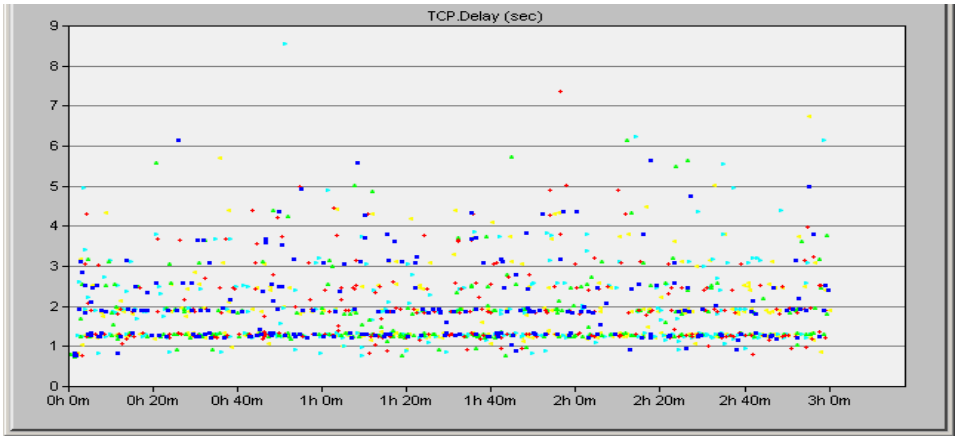

Figure 34. SINCGARS Platoon (Current Applications): TCP Delay

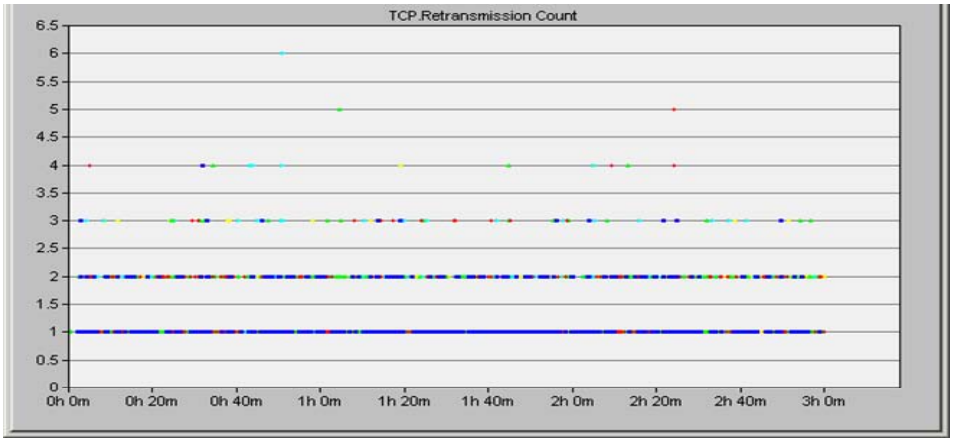

Figure 35. SINCGARS Platoon (Current Applications): TCP Retransmission Count

b. EPLRS Performance

The EPLRS network running the Current Applications scenario was only able to successfully support the Video application and failed to support both the Position Update and Short Message applications. The decrease from 94.49% to 24.24% of the Position Update success rate was enormous, and is representative of how increased network loads can drastically affect the performance of a network that does little to ensure quality of service. With the high rate of Video application messages being introduced onto the network, it is not surprising that many of the Position Update messages would collide with some number of the 21,600 video messages being broadcast during each simulation.

The decrease in success rate for the Short Message application dropping back down to 60.58% is not surprising, since in this scenario none of the other applications use the pseudo randomly generated transmission times. This is almost the exact same success rate as the Short Message Only scenario, and is likely due to the similarly timed message collisions occurring in this scenario as the Short Message Only scenario.

Under this scenario's network load, Figures 36 and 37 show that the average transmission throughput for each radio increases to almost ten times higher than the previous scenario. This demonstrates the effect relaying communication nodes can have on overall network load. As the number of generated messages across the network has increased, so does the number of message relays sent across the network, which increases the amount of total network transmissions. Since the Video application introduces a significantly greater amount of traffic, we should expect the overall network throughput to increase accordingly.

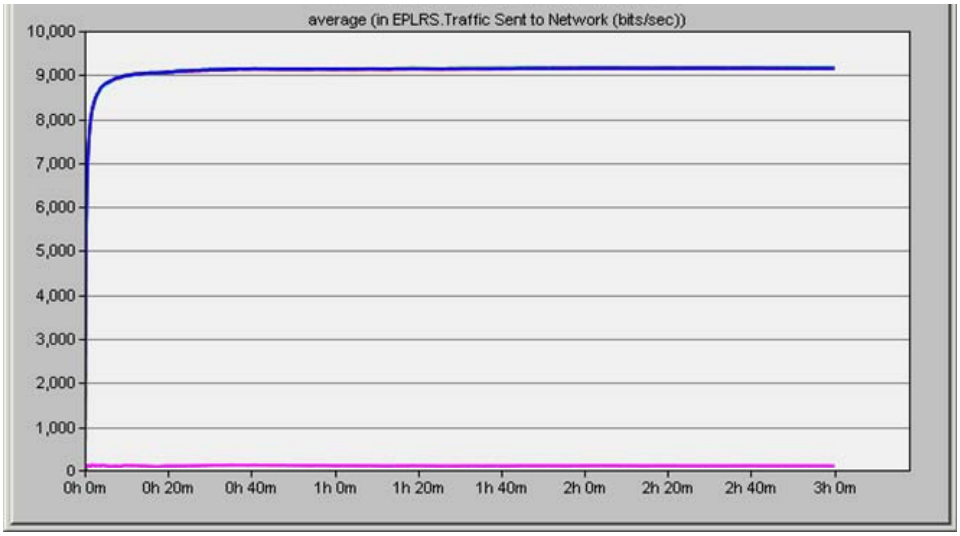

Figure 36. EPLRS Platoon (Current Applications): Average Tx Throughput

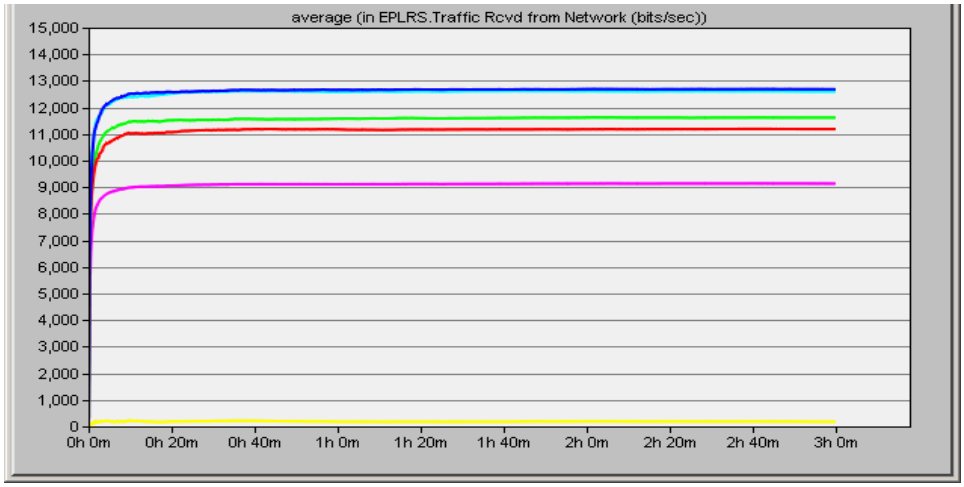

Figure 37. EPLRS Platoon (Current Applications): Average Rx Throughput

Overall, the TCP statistics reflected the impact of increased network traffic, showing longer delays and higher retransmission counts. As shown in Figure 38 and 39, the average TCP delay was 2.22 seconds, almost eight times the delay experienced by the same network in the previous scenario, and the maximum TCP delay increased from 22 seconds to 75 seconds (not considering the anomalous 6206.2 second delay created by having an infinite maximum hold time setting). The average TCP retransmission count was 2.55, only slightly higher than the previous scenario, and had a maximum TCP retransmission count of 10 retransmissions (down from 15 in the previous scenario).

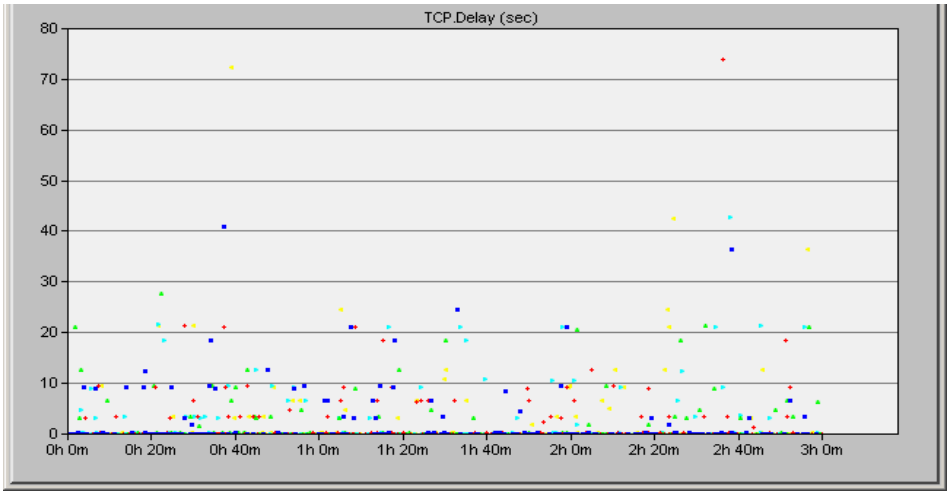

Figure 38. EPLRS Platoon (Current Applications): TCP Delay

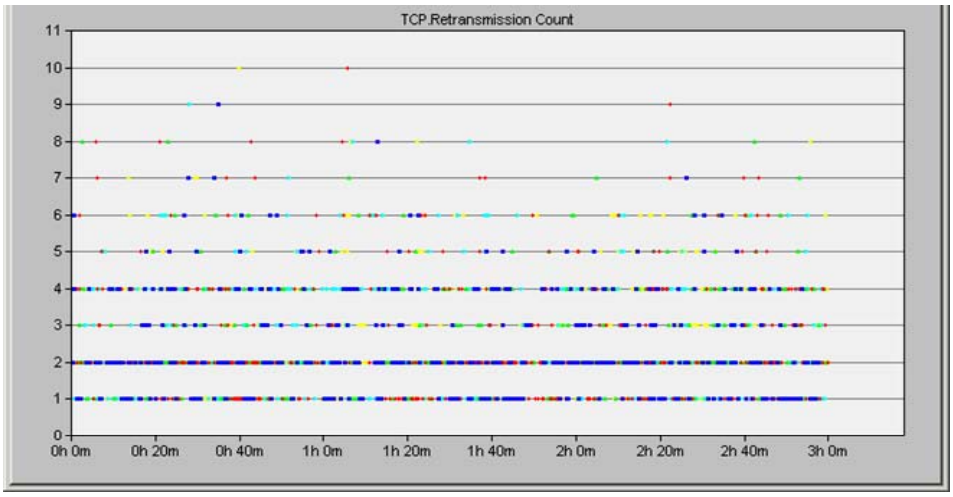

Figure 39. EPLRS Platoon (Current Applications): TCP Retransmission Count

5. All Applications

A summary of each network's performance under this scenario is shown in Table 6. While the results presented in [1] show that the CDR network failed to support the Email and HTTP applications, it significantly outperformed both the SINCGARS and EPLRS networks during the simulation of this scenario.

Overall, the SINCGARS network failed to provide simultaneous support to all of this scenario's applications. The EPLRS network gave a generally mediocre performance, as well, even though it performed noticeably better than the SINCGARS. The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

%	CDR	SINCGARS	EPLRS
Position Update	98.9	17.89	26.22
Short Message	99.6	74.2	62.87
IRC	100	8.67	95.5
HTTP	72	0.83	58.33
Email	39	0.83	58.33
Video	98.8	13.5	95.64

Table 6. All Applications Application Success Rates (Platoon).

a. SINCGARS Performance

The performance of the SINCGARS network during this simulation scenario was dismal at best. Its highest application message delivery success rate was 74.2% for the Short Message application, and it achieved a 99.2% failure rate for the delivery of all HTTP and Email application messages. With a sudden increase in network resource demand, the ability of the SINCGARS network to support even one single application plummeted. This further demonstrates the inability of a network that does not provide quality of service controls, such as CSMA or cooperative diversity, to support busy networks.

While the average transmission and reception throughputs remain relatively unchanged from the previous scenario, as displayed in Figures 40 and 41, the most noticeable differences are in the TCP performance metrics. Figure 42 shows that the average TCP delay increased from 1.96 seconds to 77.3 seconds, with a maximum delay of 3809.4 seconds. This figure shows an almost exponential increase in TCP delays, and does not show an increase in delay after about the one hour and twenty minutes point in the simulation, since the final delay values for these messages would have placed their delivery well beyond the duration of simulation run. However, the average TCP retransmission count was 1.27, surprisingly less than that of some of the previous scenarios, as shown in Figure 43.

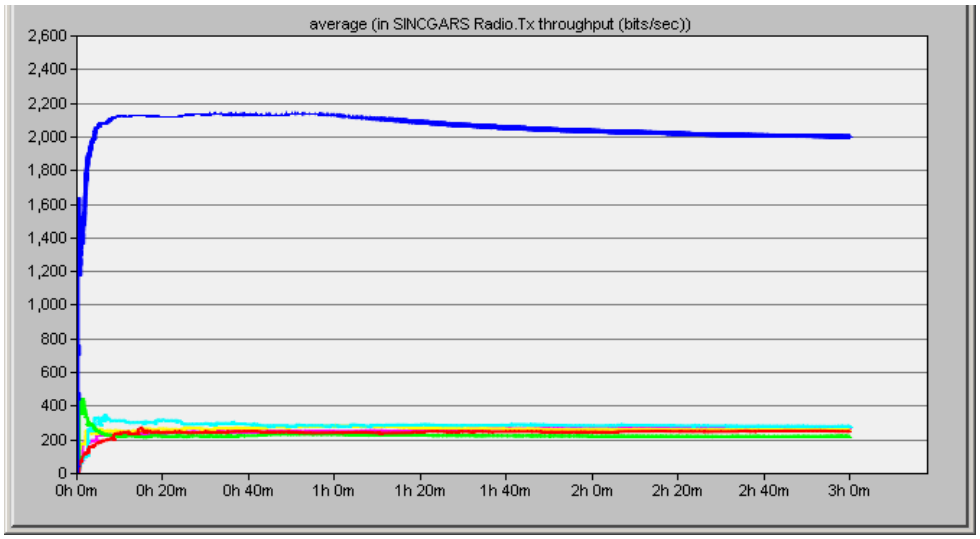

Figure 40. SINCGARS Platoon (All): Average Tx Throughput

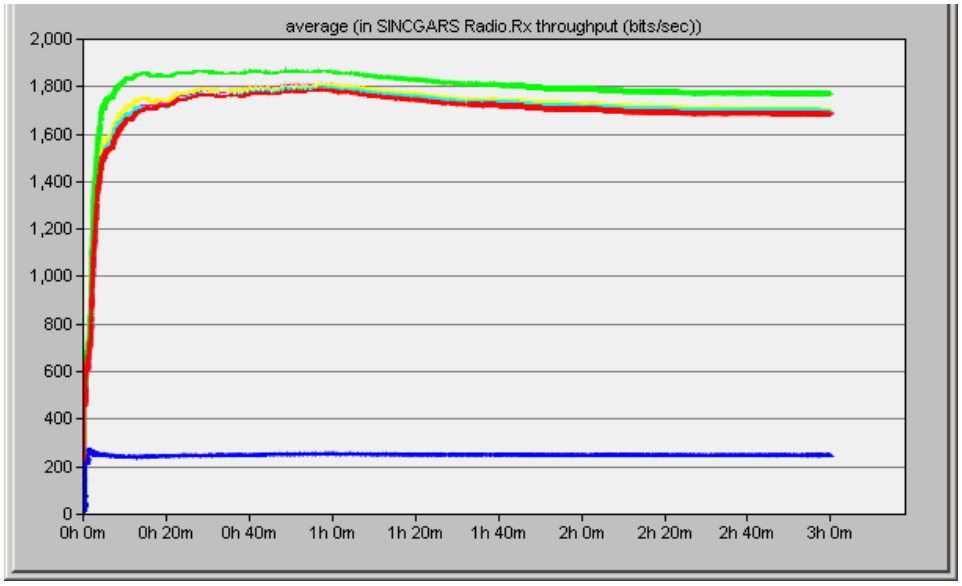

Figure 41. SINCGARS Platoon (All): Average Rx Throughput

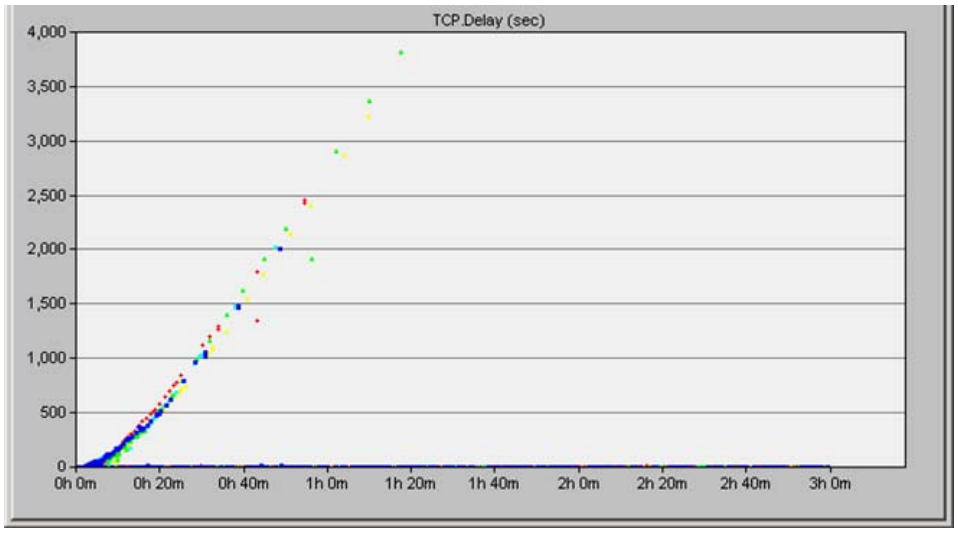

Figure 42. SINCGARS Platoon (All): TCP Delay

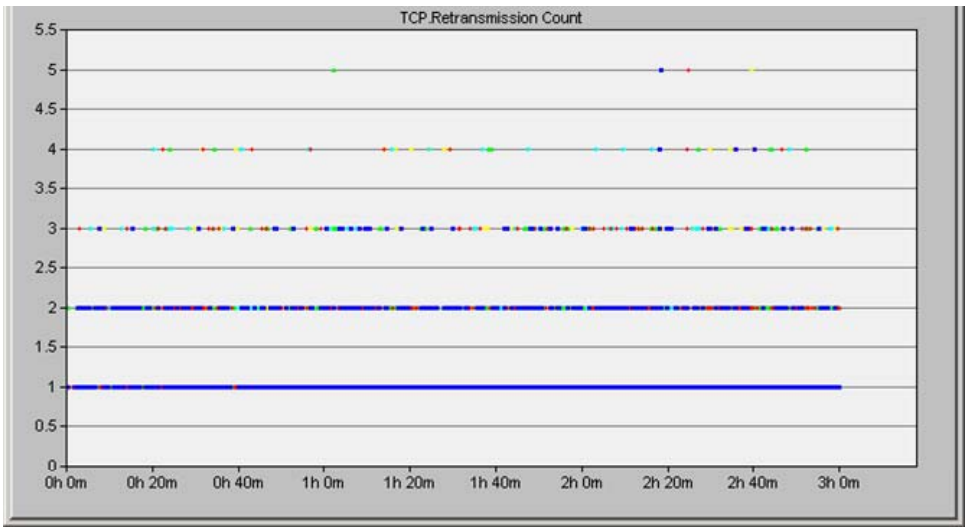

Figure 43. SINCGARS Platoon (All): TCP Retransmission Count

The SINCGARS network's inability to handle this amount of network traffic is due to the large number of transmissions colliding with each other. Since each node will transmit as soon as it has traffic to send, instead of waiting to transmit at the beginning of a time slot, it is very likely that one node will interrupt the transmission of another. In networks with high traffic volume, the use of transmission time slots, such as the Slotted-ALOHA protocol in use by the CDR model is preferred, because it can help minimize the chances of transmission interfering with each other. If each node waits to

transmit new traffic at the beginning of a time slot, then their transmissions are less likely to collide with the tail end of a message that is already being transmitted by another node. Since the SINCGARS has no method for mitigating these types of collisions, we should not be surprised at its poor performance during this simulation scenario.

b. EPLRS Performance

While the performance of the EPLRS network during this simulation scenario was better than that of the SINCARS, it also failed to adequately support all of the application traffic generated during this network load. Its highest application message delivery success rate was 95.64% for the Video application, which may seem good, but since our Video application is only transmitting video at two frames per second, such a degraded performance of an already less-than-poor-quality-video application cannot be considered a success.

While the average transmission and reception throughputs remain relatively unchanged from the previous scenario, as shown in Figures 44 and 45, the most noticeable differences are in the TCP performance metrics. Figure 46 shows that the average TCP delay decreased from 2.22 seconds to 1.11 seconds from the EPLRS results in the previous scenario (even though the maximum delay of 190.74 seconds was almost three times greater than in the previous scenario). While the average TCP retransmission count was 2.54, shown in Figure 47, roughly the same as that of some previous EPLRS scenarios, it is still more than double that of the SINCGARS network during the same scenario, further illustrating the effects of increase collisions from transmission relays.

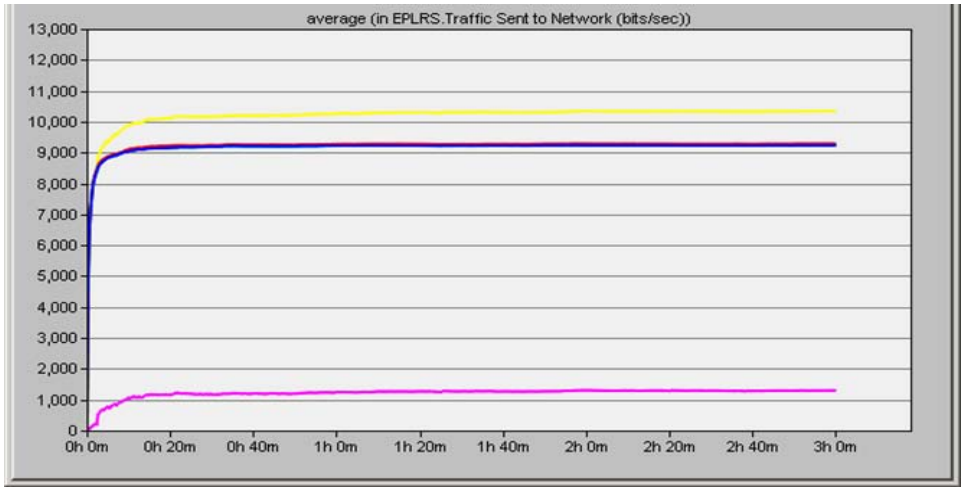

Figure 44. EPLRS Platoon (All): Average Tx Throughput

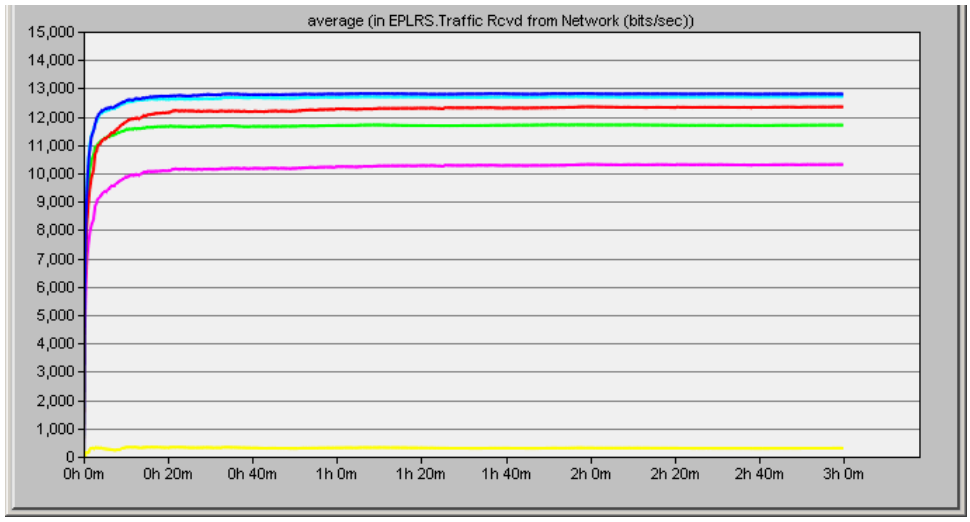

Figure 45. EPLRS Platoon (All): Average Rx Throughput

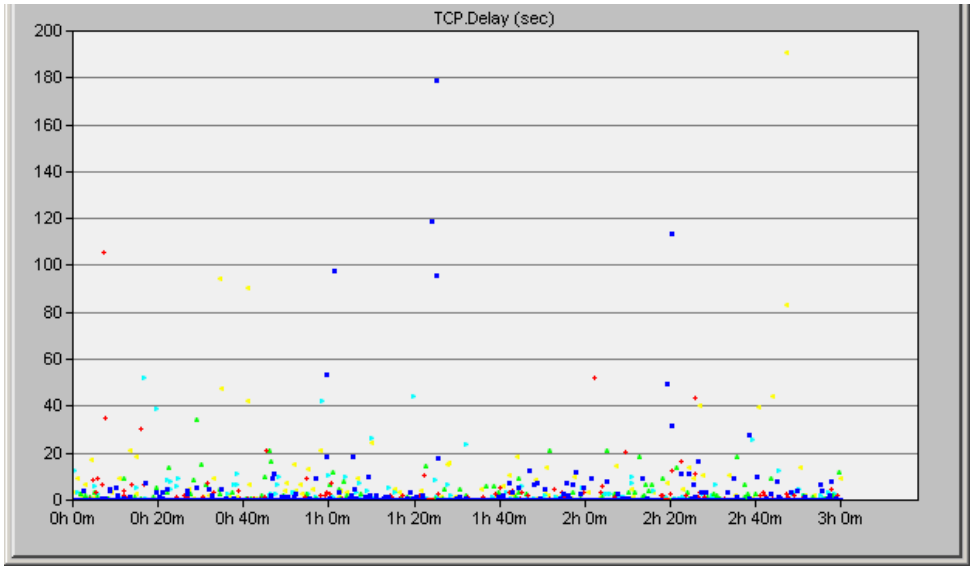

Figure 46. EPLRS Platoon (All): TCP Delay

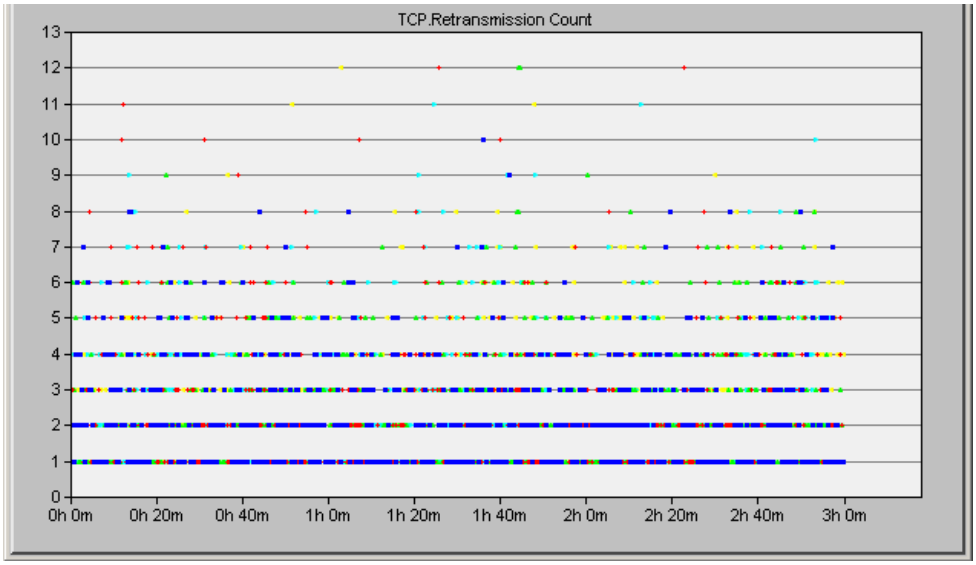

Figure 47. EPLRS Platoon (All): TCP Retransmission Count

C. COMPANY SIMULATIONS

Our company-sized network simulations consist of twenty radios: three platoons identical to the one platoon from the previous set of scenarios, one mortar platoon, one gateway node, one company executive officer, one company commander, one forward

observer, one JTAC, one UAV and one weapons platoon commander. Figure 48 depicts the location of each radio model within the simulated network.

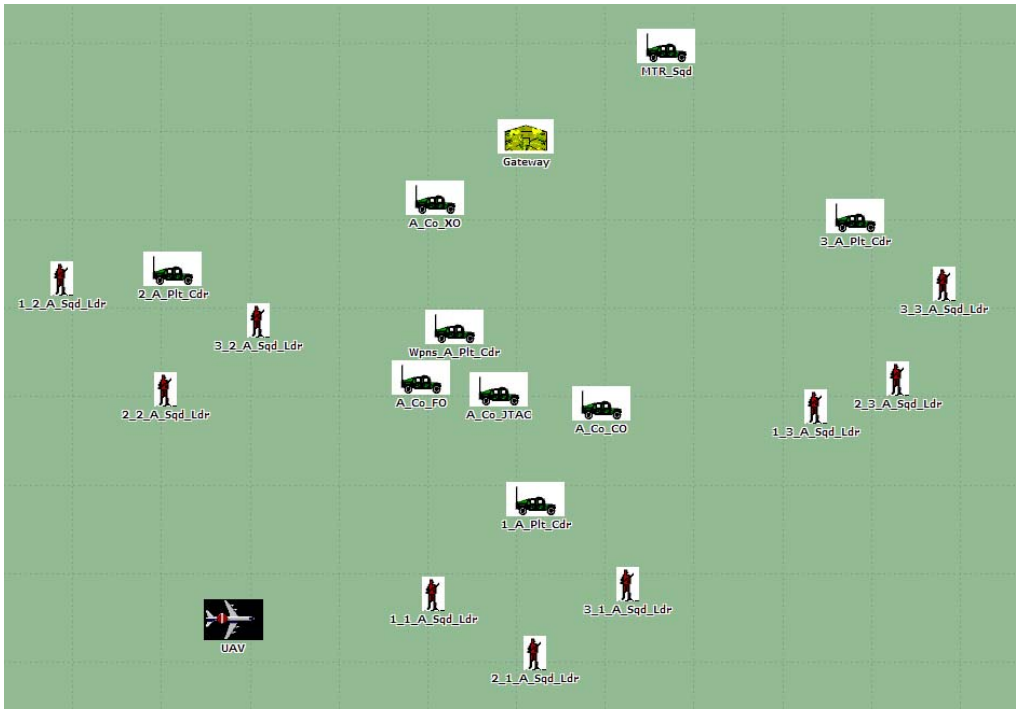

Figure 48. Company Network Layout

Each node attached to the network is limited to transmitting the types of traffic allowed by the application profile associated with that particular node during each scenario. Therefore, depending on the scenario, some nodes may introduce different types and quantities of network traffic than the other nodes. The applications associated with each node attempt to mimic the types of traffic that would be reasonably expected from the type of job assigned to each node. A squad leader is only allowed to use the position update and short message applications; the mortar platoon, forward observer and JTAC can only use the position update, short message and fire support applications; and the platoon commander, XO and CO can use all application profiles. The gateway node does not initiate the use of any applications. It only acts as a source for streaming video, a recipient of position report data and source of TCP application traffic, for nodes sending TCP traffic that is routed through the gateway.

1. Position Update Only

All three networks were able to support the application run in this simulation scenario. The results presented in [1] show that the CDR network achieved a lower message delivery success rate of 95.0% during this scenario, and because it sometimes failed to deliver three to four messages in a row for various nodes, it was graded as a failing to support the application. We believe this criterion to be somewhat harsh, since most tactical units will not have traveled a significant distance in a matter of only three or four minutes. Those units that are travelling fast enough to require higher precision position location reporting are rarely not travelling in groups, so while one node within the group may not be reporting current position information, the reports from other nodes associated with the same groups will more than make up for the lag in reporting from a single node. So, to be fair to the results shown in the CDR simulation, while we consider its support of the Position Update application in this scenario to be a success, we will upgrade it only to "Marginal" in our summary of results section at the end of this chapter. The SINCGARS network achieved a 100% message delivery success rate and the EPLRS network achieved a message delivery success rate of 97.62%. The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

a. SINCGARS Results

The SINCGARS network demonstrated the same results in the Company Position Update Only scenario as it did in the Position Update Only Platoon scenario. It achieved a 100% message delivery success rate, with a maximum receive and transmit throughput of 624 bps, and average throughputs of around 300 bps. Once again, these results are not very surprising, because each of the messages were offset from each other and transmitted at a constant rate, so with transmission overlaps caused by variations in transmission times, we expected there to be no transmission collisions. Potential collisions experienced in actual deployment of a SINCGARS implementation of such a

lightly-loaded network could be easily mitigated by an application level CSMA MAC protocol. The graphs for this scenario look exactly the same as those for the Platoon scenario, so they are not included in this section.

b. EPLRS Results

The average EPLRS node transmission throughput for this scenario was almost four times as high as the platoon version of this scenario, and the average reception throughput was roughly eight times as high, as shown in Figures 49 and 50. It is interesting to note that the SINCGARS network did not show the same overall throughput deviations from the platoon simulations, and it achieved 100% message delivery, while the EPLRS network only achieved a 97.62% message delivery success rate. This is because each SINCGARS node makes no attempt to relay other nodes' messages. This demonstrates that the use of a multi-hop network in a scenario where most nodes are within one hop of each other may actually degrade overall network performance.

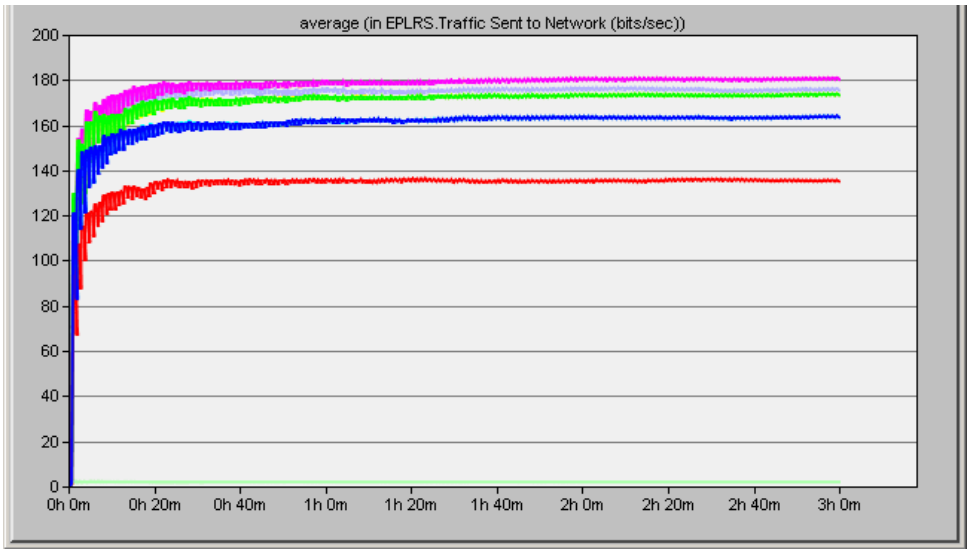

Figure 49. EPLRS Company (Position Update Only): Average Tx Throughput

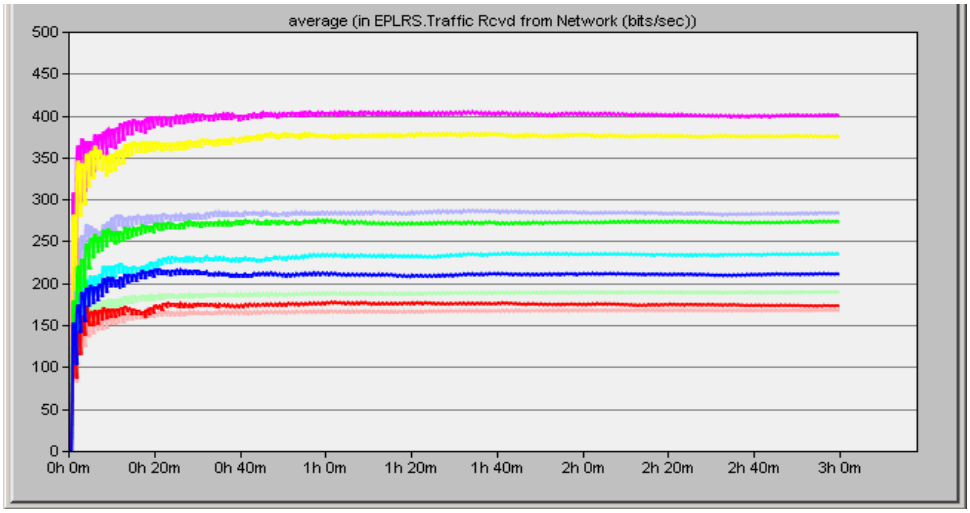

Figure 50. EPLRS Company (Position Update Only): Average Rx Throughput

2. Short Message Only

The CDR achieved a 98.6% message delivery success rate, with an average TCP latency of 1.67 seconds. This message completion rate is not considered acceptable, since text messages are rarely sent in a tactical environment that do not contain important and usually somewhat time sensitive communications. Since even a single missed message could have significant impact on operations, it may be prudent to augment the TCP acknowledgment for short message application messages with a protocol that forces an automatic retransmission of failed messages. The SINCGARS achieved a marginal 99.79% overall success rate and the EPLRS network achieved a failing 55.47% overall success rate. The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

a. SINCGARS Results

The SINCGARS network achieved a 99.79% message delivery success rate for the Short Message Only Company scenario, only slightly lower than the success rate achieved during the platoon version of the same scenario. Of the 7991 messages delivered, only 17 failed to arrive at their destination. The peak reception and transmission throughputs were 2880 bps, with an average throughput of around 280 bps.

The average node throughput graphs for this scenario look the same as those for the Platoon scenario, so they are not included here.

Figure 51 shows how both the average TCP delay and maximum TCP delay showed only slight increases from the platoon version of this scenario, increasing from 0.772 and 2.082 seconds to 0.8625 and 3.4 seconds, respectively. The average and peak TCP retransmission counts increased from the platoon scenario, with an average retransmission count of 1.8077 (up from 1.25), and a maximum count of 9 retransmissions (up from only 2), as depicted in Figure 52, and is a result of the increased quantity of overall transmissions being broadcast across this network.

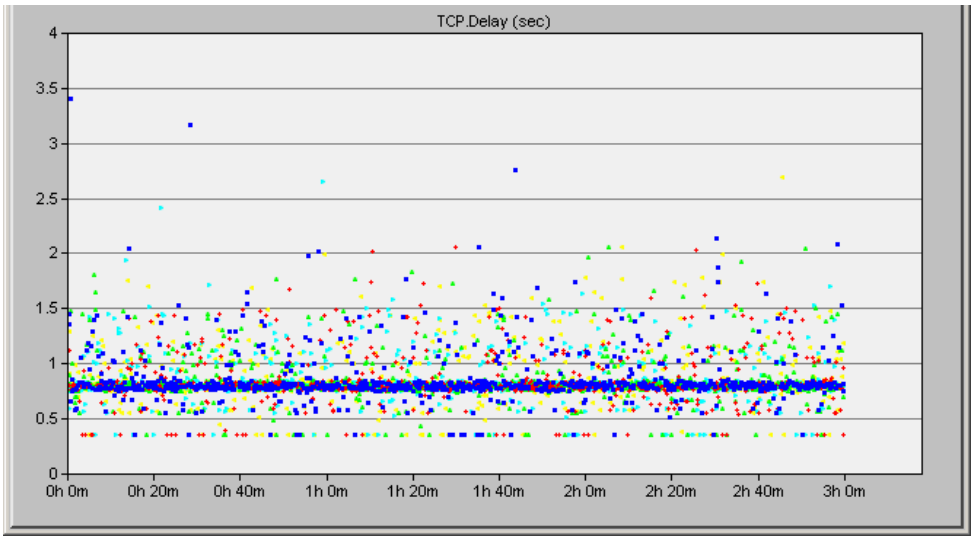

Figure 51. SINCGARS Company (Short Message Only): TCP Delay

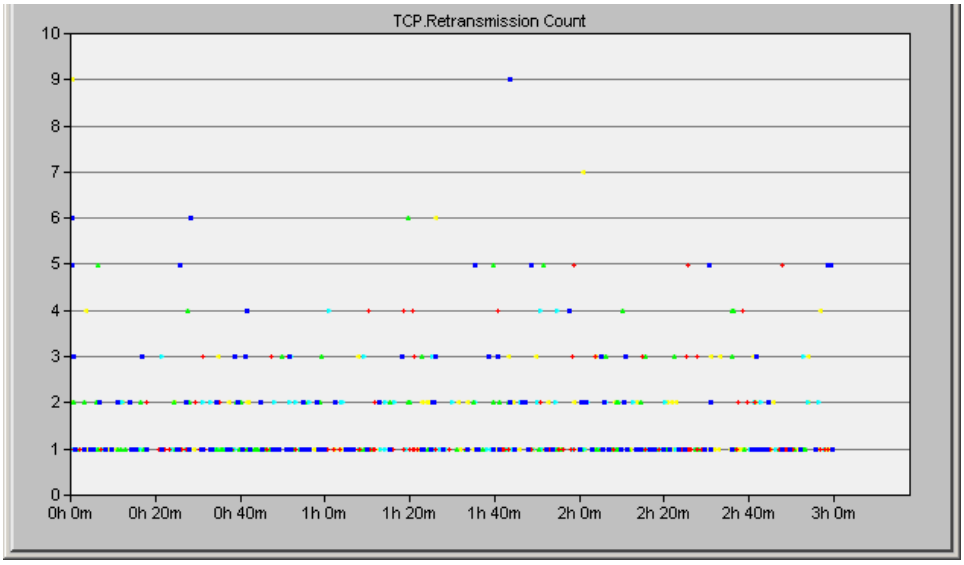

Figure 52. SINCGARS Company (Short Message Only): TCP Retransmission Count

b. EPLRS Results

In Figures 53 and 54, we see that some of the nodes closest to the middle of the network (JTAC and CO nodes – top two lines) both receive and transmit more traffic than those nodes that are further away from the other nodes (MTR node – bottom line). This is because the nodes in the middle of the network are attempting to relay traffic from one side of the network to the other, which is beneficial if the fringe nodes are not within transmission range of each other, but not when the relays are unnecessary. We also notice that the amount of transmission and reception throughputs are roughly nine times greater here, than they were during the platoon simulations runs, whereas the SINCGARS network showed almost no throughput increases. This further illustrates the significant increases in overall network traffic experienced when using wireless multi-hop technologies, and should show why these types of networks may not always be the most desirable configuration for every type of network load. While multi-hop networks are intended to extend the reachability of mobile (and fixed) nodes, the characteristics inherent to multi-hop transmissions can degrade network performance if all nodes are within transmission range of each other. Unless there is a way for groups of nodes to perceive their proximity to each other and dynamically alter their multi-hop

configurations, applications that function well when all nodes are spread out, may not perform as well when they become more tightly grouped.

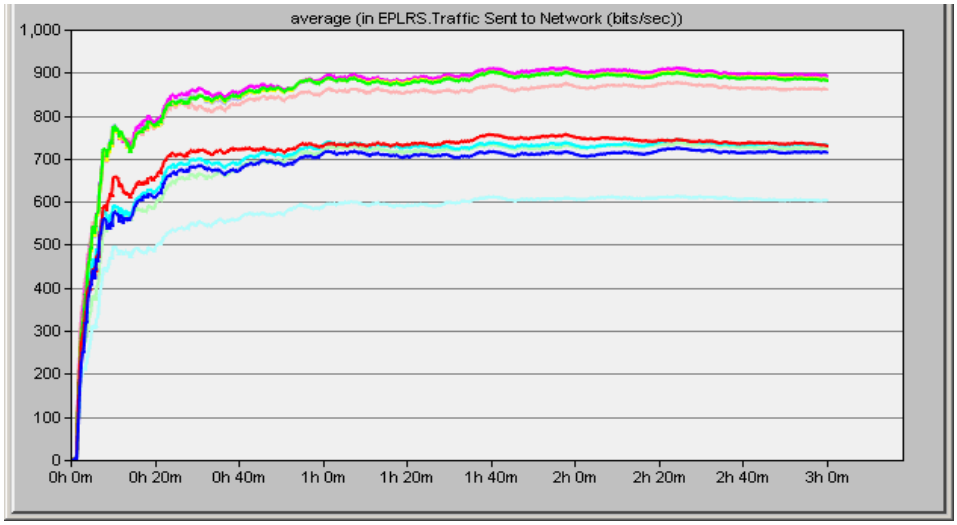

Figure 53. EPLRS Company (Short Message Only): Average Tx Throughput

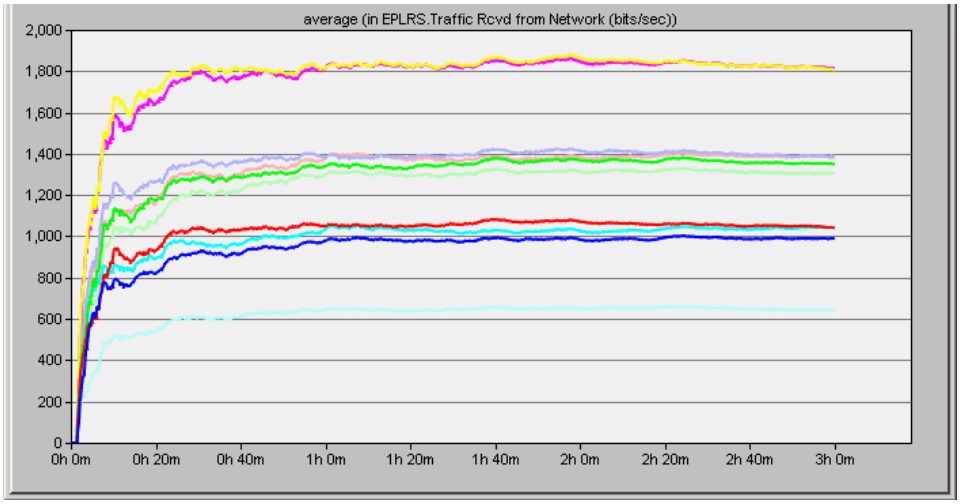

Figure 54. EPLRS Company (Short Message Only): Average Rx Throughput

Figure 55 shows that the average TCP Delay was 1.393 seconds, with a maximum delay of 92.3 seconds, which is significantly greater than the 0.862 second average and 3.4 second peak TCP delays experienced by the SINCGARS network. As depicted in Figure 56, the average TCP retransmission count was 6.736, with a maximum of 25, which is more than double the retransmission counts experienced by the EPLRS network running the same applications on the platoon scenario.

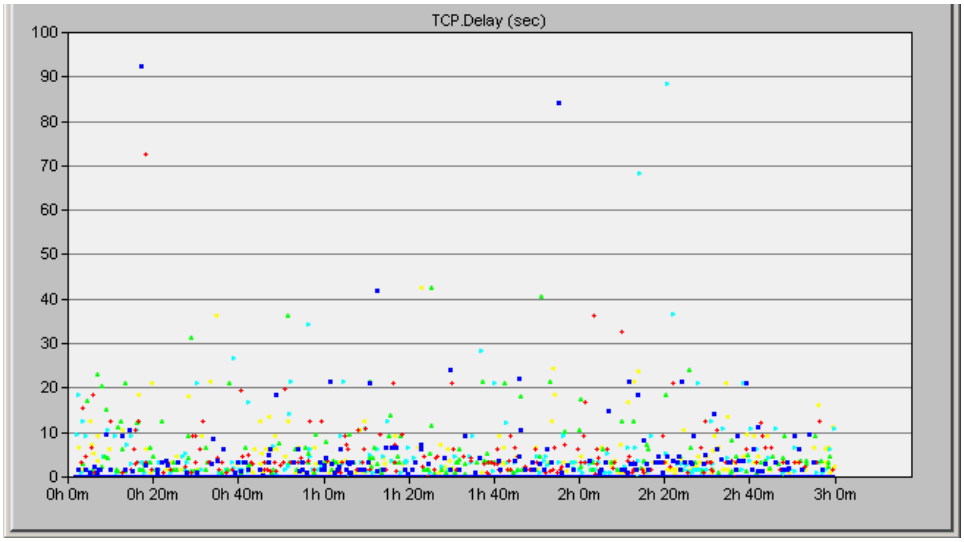

Figure 55. EPLRS Company (Short Message Only): TCP Delay

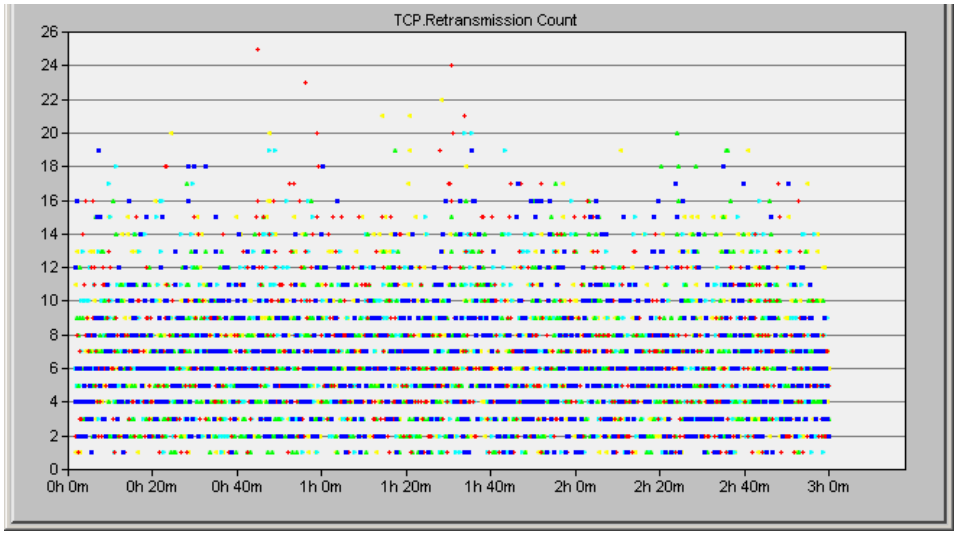

Figure 56. EPLRS Company (Short Message Only): TCP Retransmission Count

3. Commanders and Position

As explained in [1], the CDR simulation runs were only able to complete between 12-60 minutes of each 3-hour simulation. No CDR network message completion rates were provided for this scenario, but it was mentioned that the HTTP and Email applications failed to deliver all of their application messages. The SINCGARS network achieved a 24.4% overall message delivery success rate and the EPLRS network

achieved a 66.59% overall message delivery success rate. The results of the SINCGARS and EPLRS networks area presented in greater detail in the following two sections.

%	CDR	SINCGARS	EPLRS
Position Update	Not Available	100	0
Short Message	Not Available	92.77	66.49
IRC	Not Available	1.81	82.94
HTTP	0	82.65	93.33
Email	0	82.65	93.33

Table 7. Commanders and Position Application Success Rates (Company)

a. SINCGARS Results

The initial run of this simulation scenario crashed after 1 hour and 49 minutes of simulation time, due to the JCSS program running out of memory to allocate to support all of the simulation events. Consequently, we were only able to successfully complete five 1-hour simulations. Despite the reduced simulation time, enough data was collected to draw some useful conclusions regarding the performance of the SINCGARS network under the Commanders and Position simulation scenario.

The SINCGARS network achieved a significantly lower overall message delivery success rate during this run than it did during the platoon version of the same scenario. Its overall message delivery success rate was only 24.4% for all transmissions generated during this scenario. This is not entirely surprising, since the higher number of nodes introducing traffic was expected to cause a higher rate of transmission collisions. The overall success rate seems low when compared to the individual application success rates, but since some applications send significantly more messages than others, the overall message delivery success rate can be impacted greatly by the failure of only the position update and IRC messages, because these applications generate the most amount of network traffic. This simulation run illustrates how the non-CSMA MAC protocol being used with the SINCGARS network does not perform adequately under high network traffic load scenarios.

The peak reception and transmission throughputs were 12000 bps, with an average throughput generally around 300 bps. As show in Figure 57, the average reception throughput was roughly half that of the same application set run under the platoon scenario. This lower reception throughput is due to the increased amounts of transmission collisions. If a transmission is received and categorized as noise, it is not factored into the overall reception throughput calculations. The graphs for this scenario look the same as those for the Platoon scenario, so they are not included here.

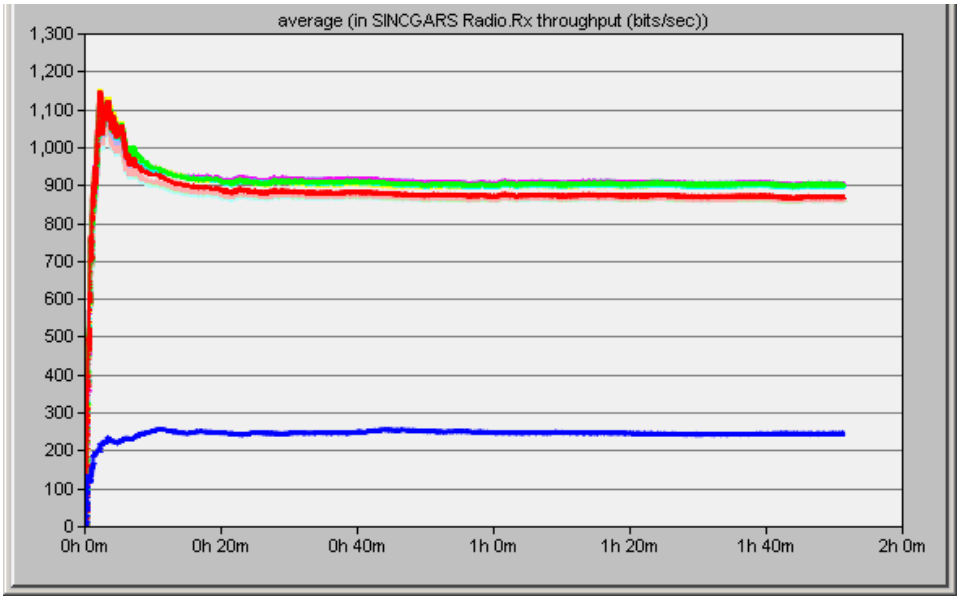

Figure 57. SINCGARS Company (Commanders & Position): Average Rx Throughput

Figure 58 shows that both the average TCP delay and maximum TCP delays shifted drastically from the platoon version of this scenario, increasing from 1.12 and 11.1 seconds to 40.79 and 199.6 seconds, respectively. This is another significant illustration of how providing no quality of service for a higher traffic density network will result in extremely poor performance. The average and peak TCP retransmission counts showed much less significant increases from the platoon scenario, with an average retransmission count of 1.867 (up from 1.12), and a maximum count of 11 retransmissions (up from only 2), as depicted in Figure 59.

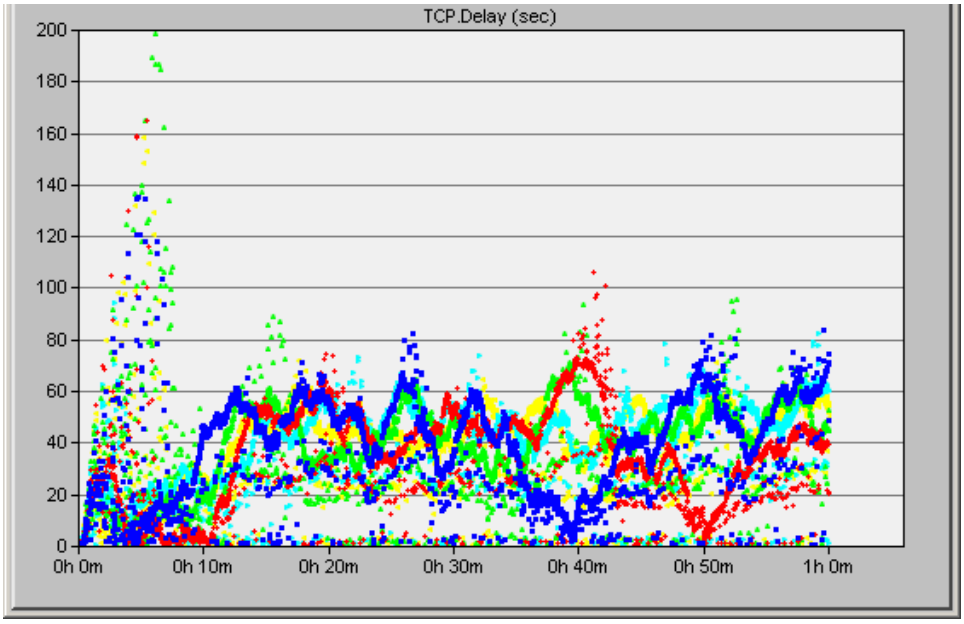

Figure 58. SINCGARS Company (Commanders & Position): TCP Delay

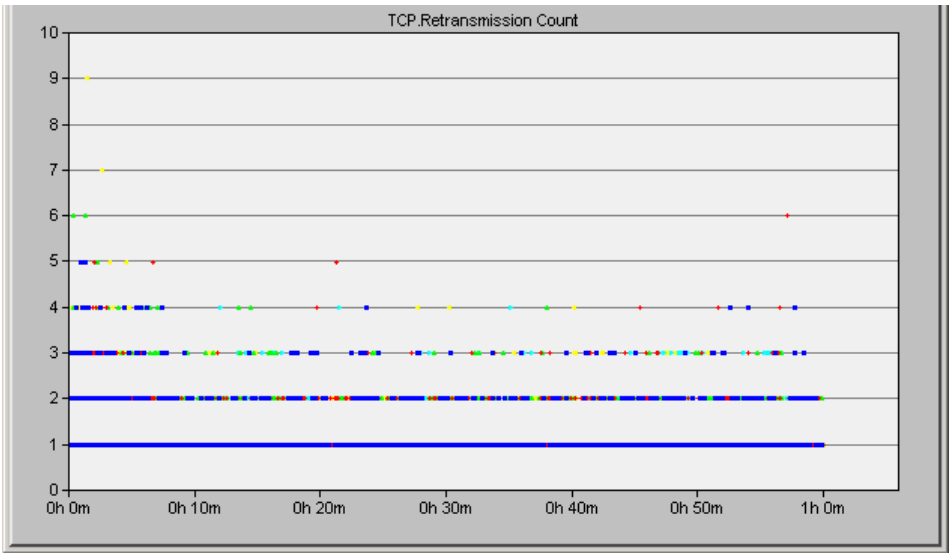

Figure 59. SINCGARS Company (Commanders & Position): TCP Retransmission Count

b. EPLRS Results

The EPLRS network achieved a significantly lower overall message delivery success rate during this run than it did during the platoon version of the same scenario. This is not entirely surprising, since the higher number of nodes introducing traffic was expected to cause a higher rate of transmission collisions. The EPLRS

achieved slightly better performance over the SINCGARS network for this scenario, which demonstrates the CSMA node's advantage over a non-CSMA node for busier networks.

Figures 60 and 61 show that the average transmission and reception throughputs are enormously larger than both the company-sized commanders and position SINCGARS throughputs, and the platoon-sized commanders and position EPLRS throughputs (as much as 40 times larger for some nodes). This increase in throughput is caused by both the increase in traffic from there being more nodes initiating traffic, and more relaying transmissions associated with each additional node.

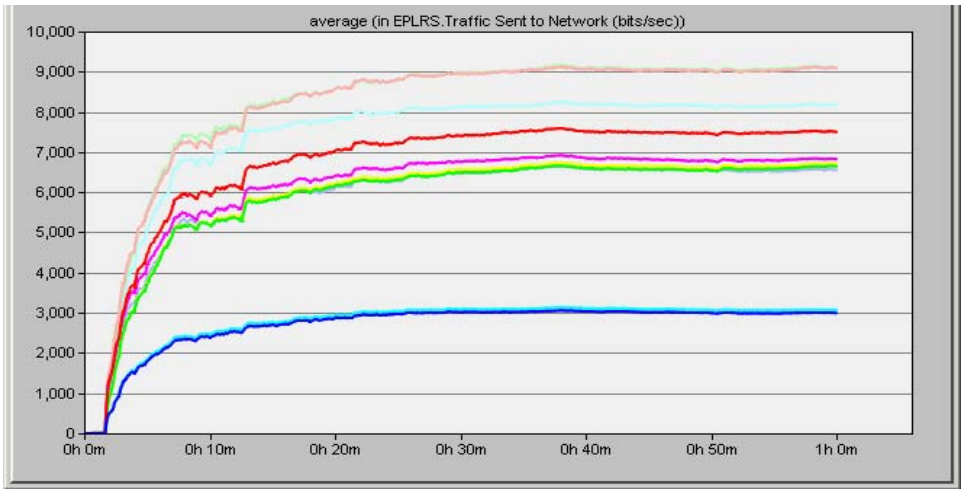

Figure 60. EPLRS Company (Commanders & Position): Average Tx Throughput

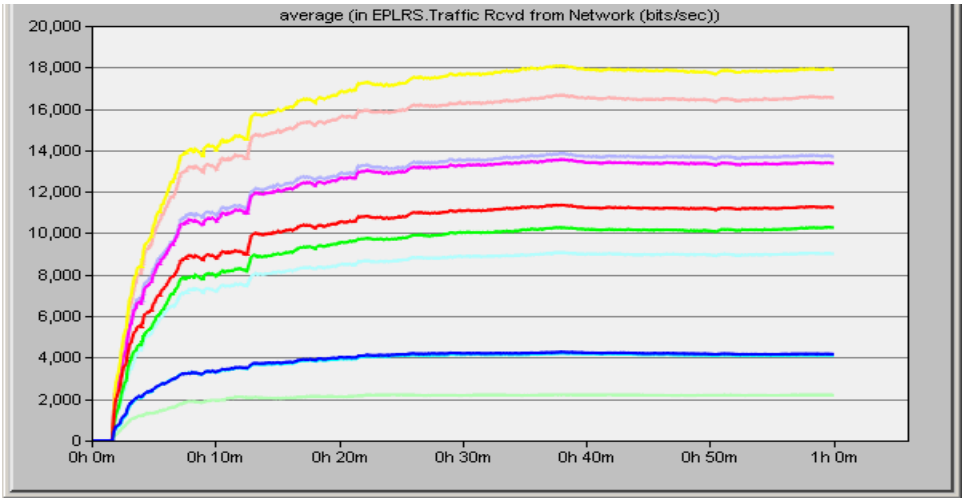

Figure 61. EPLRS Company (Commanders & Position): Average Rx Throughput

Figure 62 shows that both the average TCP delay and maximum TCP delays increased from the platoon version of this scenario, increasing from 2.24 and 15 seconds to 2.66 and 433.6 seconds, respectively. This is another significant illustration of how greatly message relaying can affect the performance network. The average and peak TCP retransmission counts showed much less significant shifts from the platoon scenario, with an average retransmission count of 4.143 (up from 2.24), and a maximum count of 15 retransmissions (down from 21), as depicted in Figure 63.

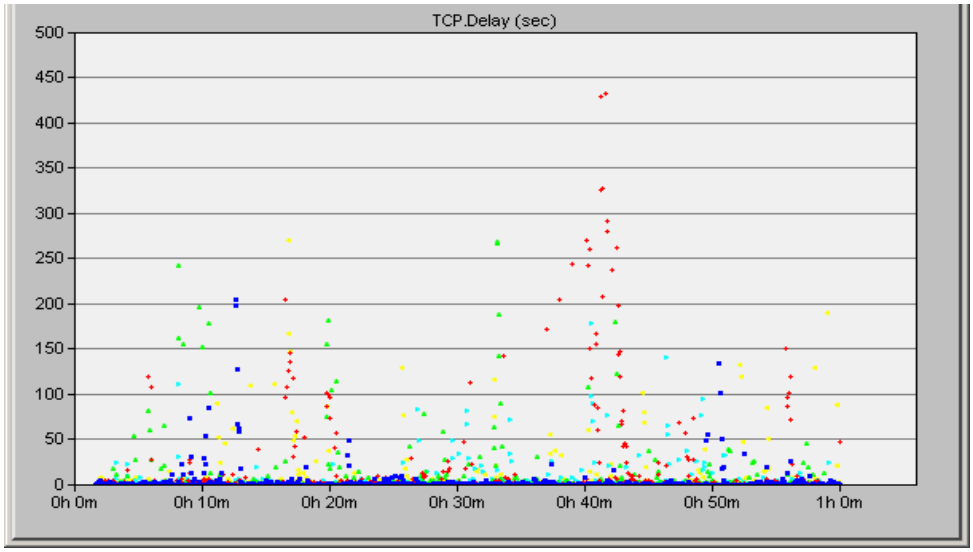

Figure 62. EPLRS Company (Commanders & Position): TCP Delay

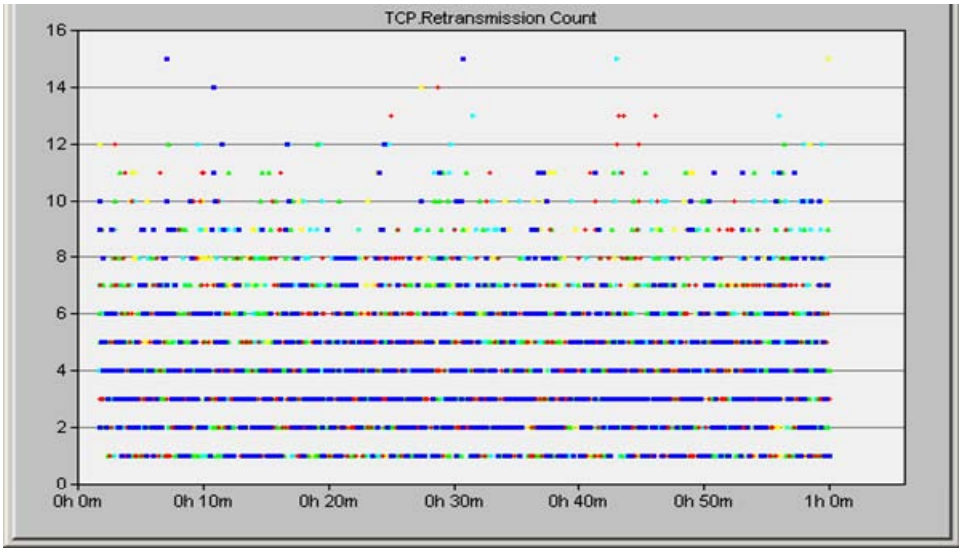

Figure 63. EPLRS Company (Commanders & Position): TCP Retransmission Count

D. SUMMARY OF RESULTS

Our network simulation results for the SINCGARS and EPLRS radio networks are presented in Tables 8 and 9. We have categorized each radio's performance in each scenario as achieving one of three levels of support: Successful, Marginal, and Failure. The Position Update application is graded less strictly, as the nature of these applications allow for greater tolerance of undelivered messages, and the Short Message and Email applications are graded more strictly, since the failure to deliver one of these messages can result in much more serious consequences. Since the Video application represented an already poor quality video feed of two frames per second, a network's inability to provide near 100% support to this resulted in a failing grade. If a network cannot support this degraded version of our Video application, then it would certainly not be able to support the greater demands of a regular quality video feed.

CDR (Platoon)					
	Position Update Only	Short Msg Only	Cdr & Pos	Current	All
Position Update	SUCCESS		SUCCESS	SUCCESS	SUCCESS
Short Message		MARGINAL	SUCCESS	MARGINAL	MARGINAL
IRC			SUCCESS		SUCCESS
HTTP			MARGINAL		FAILURE
Email			MARGINAL		FAILURE
Video				MARGINAL	MARGINAL
SINCGARS (Platoon)					
	Position Update Only	Short Msg Only	Cdr & Pos	Current	All
Position Update	SUCCESS		SUCCESS	SUCCESS	FAILURE
Short Message		MARGINAL	MARGINAL	FAILURE	FAILURE
IRC			SUCCESS		FAILURE
HTTP			SUCCESS		FAILURE
Email			SUCCESS		FAILURE
Video				FAILURE	FAILURE
EPLRS (Platoon)					
	Position Update Only	Short Msg Only	Cdr & Pos	Current	All
Position Update	SUCCESS		SUCCESS	FAILURE	FAILURE
Short Message		FAILURE	FAILURE	FAILURE	FAILURE
IRC			FAILURE		MARGINAL
HTTP			MARGINAL		MARGINAL
Email			MARGINAL		MARGINAL
Video				MARGINAL	FAILURE

Table 8. Network Simulation Results (Platoon)

CDR (Company)					
	Position Update Only	Short Msg Only	Cdr & Pos	Current	All
Position Update	MARGINAL		FAILURE	N/A	N/A
Short Message		FAILURE	FAILURE	N/A	N/A
IRC			FAILURE		N/A
HTTP			FAILURE		N/A
Email			FAILURE		N/A
Video				N/A	N/A
SINCGARS (Company)					
	Position Update Only	Short Msg Only	Cdr & Pos	Current	All
Position Update	SUCCESS		SUCCESS	N/A	N/A
Short Message		MARGINAL	FAILURE	N/A	N/A
IRC			FAILURE		N/A
HTTP			FAILURE		N/A
Email			FAILURE		N/A
Video				N/A	N/A
EPLRS (Company)					
	Position Update Only	Short Msg Only	Cdr & Pos	Current	All
Position Update	MARGINAL		FAILURE	N/A	N/A
Short Message		FAILURE	FAILURE	N/A	N/A
IRC			FAILURE		N/A
HTTP			MARGINAL		N/A
Email			MARGINAL		N/A
Video				N/A	N/A

Table 9. Network Simulation Results (Platoon)

From these results, we can conclude that under very light network loads, the SINCGARS radio network, with its ALOHA-like MAC protocol, performed better under lighter network loads, but as the amount of network traffic increased, its performance quickly deteriorated.

In light of its occasionally better performance for some of our simulation scenarios, it is worth pointing out that the SINCGARS does not support multi-hop networks. Since these scenarios only presented application traffic generated by stationary entities that were all within a single hop of each other, these results do not accurately reflect this radio's inability to support these same application requirements to mobile users who travel further than one hop away from any traffic-generating node.

Across the entire set of our network simulations, the EPLRS device was only able to support 2 of the 23 application instances successfully. It is worth noting that, while this device does support up to 32 simultaneously separated channels (or needlines), we chose to have all applications in each of our simulations share the same channel, in order to provide a more accurate basis for comparison of the performance of a single EPLRS needline to the other two radios. Since the CDR simulations in [1] attempted to support all of its applications using only a single channel, and the SINCGARS only supports the use of one channel at a time, we chose to model the EPLRS performance across a single needline, as well. Even with the greater throughput capabilities of the EPLRS device, the addition of the TCP overhead proved to be too great for this multi-hop device to support with its CSMA needline. These results do not reflect the EPLRS device's overall ability to support data applications in general, but merely demonstrate the performance characteristic differences inherent to each type of wireless network technology across a single channel.

The dismal results of our EPLRS node's use of TCP traffic demonstrates the potentially negative effects of using TCP-like protocols across multi-hop networks. When mobile multi-hop network nodes transition from being spread out to being within close proximity to each other, the increased traffic load of TCP-like protocols and the subsequent transmission relays of each of these additional transmissions can present performance degradations that hamper effective network support of tactical data applications. Our results imply that a multi-hop network implementation that assumes all nodes within a given network will maintain some degree of separation, and does not have a well-defined transmission relay policy, may not effectively support the requirements of tactical mobile data networks.

Our simulation results also demonstrate how the use of cooperative diversity by the CDR multi-hop network, when compared to the EPLRS multi-hop network that did not use cooperative diversity, effectively mitigated some of the effects of the multi-path conditions caused by message relaying in the smaller network scenarios, but was unable to effectively mitigate the same conditions for larger networks. This demonstrates that the use of cooperative diversity alone may not be able to overcome the increased

transmission load for multi-hop networks that are temporarily operating in close proximity to a large number of other multi-hop nodes belonging to the same network.

We must also point out that none of these simulations included the effects that tactical voice communications may have on creating additional network delays. While the EPLRS may be able to provide non-conflicting support for tactical voice applications with the implementation of a VoIP-only data needline, the addition of voice communication demands to the SINCGARS network would almost certainly prevent it from achieving the same level of success it had in our simulations, since all data communications would be blocked during the transmission of voice network traffic.

Additionally, these application profiles may not reflect exact parameters of similarly named applications running across actual deployed systems. These application parameters were merely used to provide reasonable comparisons to the previously determined CDR simulation results, to show comparable SINCGARS and EPLRS network performances under like network loads, and our results are not intended to imply that actual applications with similar names or purposes will perform at the same levels on the devices mentioned in this thesis. Refining the application profiles to more accurately match the characteristics of actually deployed applications will be left for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS AND FUTURE RESEARCH

A. CONCLUSIONS

With the sudden growth in the development of more efficient and more capable wireless communication technologies, it is no surprise that there are many potentially useful military applications for most of these communication advances. The military has always relied on its ability to harness the newest types of technology to enhance the effectiveness of its command and control architecture, yet much of its currently deployed tactical communication equipment predate even the widespread use of the Internet. Adaptations to these older radios have been periodically introduced, in an effort to temporarily close the gap between the older technologies and newer ones, but rarely arrive in the form of acceptable long-term technology solutions.

Newer devices have been fielded, but none of these newer devices has been capable of completely replacing our older equipment. As a result, we have a command and control architecture that is composed of many different types of networking devices that perform very specific functions very well, but are incapable of communicating with each other. With our increased demand and dependence on these newer data networking technologies, the consolidation of communication capabilities into a smaller arsenal of more advanced networking devices is imperative to the continued improvement of our tactical networking architecture.

In this thesis we recreated the network simulation scenarios used for the evaluation of an experimental cooperative diversity-based radio (CDR), analyzed in previous thesis work [1]. We used these same scenarios to evaluate communication devices that are currently in use by the military, and quantify the actual level of inability of these devices to provide the type of combined communication capabilities needed in our present day tactical environment. We subjected each device to the same sets of network traffic loads as the CDR, in order to demonstrate the actual measured shift in performance that this one new technology provides over the older communication devices. Even though the CDR implements one relatively new wireless technology, both

the SINCGARS and the EPLRS networks occasionally outperformed the CDR network with their older technologies. The primary take-away from our simulation results is that the technology being used for a particular type of communication network must match the needs of the network. If it does not do this, then overall network performance may be significantly degraded, regardless of how new or exciting the implemented technology. We saw that under lightly trafficked networks, the more complicated technologies failed to perform at the same high levels as the more simple technologies. However, as the scenarios increased their traffic load, the more complicated devices began to show more superior performance levels than the more simple communication devices.

Our simulation results illustrate some of the problems encountered when using different types of wireless networking technologies to support a variety of tactical network traffic scenarios, especially when multi-hop configurations are used in scenarios where it is not needed. We demonstrated that when all multi-hop network nodes are operating within a single hop of each other (a scenario that would not be uncommon for mobile tactical nodes), without a well-defined transmission routing or forwarding policies, these multi-hop technologies actually degrade overall network performance. Additionally, the only advantage of the cooperative diversity capability of the CDR is in its ability to simultaneously relay a single common frame by multiple sources, which implicitly requires a true multi-hop, or at least a multi-path environment to prove beneficial and also does not significantly benefit the performance of these mobile multi-hop networks under close node proximity situations. Therefore, we see that while certain emerging wireless technologies may solve connectivity issues involving communication between distant nodes, they introduce new challenges for ensuring continued communication between non-distant nodes belonging to the same network. While no one technology is going to solve all of the problems encountered in the employment of wireless mobile communication devices, the combination of the right technologies may effectively mitigate many of these issues.

B. FUTURE WORK

1. Tactical Network Application Refinement

The applications used in our JCSS simulations were recreated to mimic those presented in [1], in order to create an accurate basis of comparison for both the SINCGARS and EPLRS to the performances of the CDR Actual characteristics of applications running in current tactical environment may vary from those presented in this evaluation. The collection and validation of actual application statistics and the subsequent transformation of these statistics into JCSS application profiles could be immensely beneficial to future JCSS evaluations of tactical communication technologies, as a ready-made framework for consistent software evaluation of tactical mobile wireless devices is not currently available through OPNET or JCSS network simulation suites.

2. Wireless Mobile Device Benchmark Criteria

The combination of a standard set of JCSS/OPNET device applications with an extensive set of realistic evaluation scenarios could be used as a much more thorough benchmarking method for the comparison of currently deployed devices to experimental devices, than the scenarios presented in this thesis. Establishing device software simulation benchmarking criteria would not only present a consistent method of identifying the strengths and weaknesses of various communication devices, it would present potential vendors with a comprehensive set of criteria needed to create an "ideal radio" device for various types of applications (i.e., a device that could achieve 100 % message delivery success rate during each of the benchmark scenarios). This would have to be a large and varied set of scenarios, including simulated node trajectories, terrain modeling and various types of non-node signal interferences (environmental and hostile).

3. JCSS Scenario Refinement for Currently Deployed Devices

The creation of more complex JCSS simulation scenarios for the SINCGARS and the EPLRS would provide additional baseline comparisons for the evaluation of emerging wireless technologies. These simulations could combine larger networks of

mobile nodes that actually travel along various types and combinations of node trajectories, or combine different nodes with a variety of device configurations (i.e., changing needline type for an EPLRS network). This would provide useful insight into the actual capabilities and limitations of currently deployed communication devices, and would provide more effective baselines for further comparisons between these devices and experimental ones.

4. SINCGARS Mobile Ad Hoc Application JCSS Evaluation

Previous thesis research [6] provided an experimental implementation of an application that created MANETs using SINCGARS radios. Creating an OPNET Modeler model for this application, and then evaluating its performance under a variety of network simulation scenarios would be useful in discussing the performance of this application on a larger scale, and could be used to make comparisons to other emerging MANET technologies.

5. Multi-hop Network Protocol Analysis and Refinement

As new mobile wireless networking technologies continue to develop, it may be possible to redefine both older MAC and message exchange protocols to make more efficient use of the new technologies within a tactical environment. As demonstrated in this thesis, the increased transmission load of TCP-like protocols can introduce enough overhead to significantly degrade the performance of multi-hop networks, when most nodes in the network are within a single hop of each other. Protocols that allow for dynamic multi-hop configuration alterations, based on proximity to other nodes in the network could allow greater efficiency for networks where mobile users operate across a wide range of proximity to other nodes in the network. A methodical performance evaluation and comparison of existing wireless networking protocols with protocols that have been suggested but not yet implemented, using network simulation software would provide valuable quantification of actual improvements or degradations caused by implementing one protocol over another.

LIST OF REFERENCES

[1] Byron R. Harder, *Analytical and Simulation-Based Assessment of a Prototype Tactical Multi-Hop Radio Network*, Naval Postgraduate School, September 2008.

[2] Lang Tong, Wireless *Ad-Hoc Networks with Receiver Multipacket Reception*, Army Research Office, September 2004.

[3] Dr. Bhaskar Krishnamachari, *Scalability Analysis of the TrellisWare Approach to Ad Hoc Wireless Networking*, University of Southern California, April 2007.

[4] Adam Blair, Thomas Brown, Keith M. Chugg, Mark Johnson, *Tactical Mobile Mesh Network System Design*, TrellisWare Technologies, Inc. and University of Southern California, June 2008.

[5] *OPNET Modeler 14.5 Product Documentation*, OPNET Technologies, Inc., Bethesda, MD © 1987-2008

[6] Steven Brand, A *Software-Based Network Infrastructure For Mobile Ad Hoc Data Networking In Support Of Small Tactical Units Using the SINCGARS Radio*, Master's Thesis, Naval Postgraduate School, March 2006.

[7] *Planner's Manual for EPLRS Networks*, TB 11-5825-298-10-3, ENM Software Version 4.4, EPLRS Radio Software Version 11.4, Headquarters, Department of the Army, August 15, 2004.

[8] *SINCGARS ICOM Ground Radios*, TM 11-5820-890-10-6, Headquarters, Department of the Army, August 1, 1998.

[9] J. Nicholas Laneman, David N. C. Tse, Gregory W. Wornell, "Cooperative Diversity in Wireless Networks: Efficient Protocols and Outage Behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, December 2004.

[10] Robert D. Martin, Dr. Harlan B. Russell, *Investigation of Least Resistance Routing in a Mobile SINCGARS Packet Radio Network*, Clemson University and Techno-Sciences Inc., 1996

[11] *TALK-II SINCGARS: Multiservice Communications Procedures for the Single Channel Ground and Airborne Radio System*, FM 11-1, Air Land Sea Application Center, May 1996.

[12] *Enhanced Position Location Reporting System (EPLRS) Model User's Guide*, OPNET Technologies, Inc., Bethesda, MD © January 7, 2008.

[13] *TACTICAL RADIOS: Multiservice Communications Procedures for Tactical Radios in a Joint Environment*, FM 6-02.72, Air Land Sea Application Center, June 2002.

[14] James F. Kurose, Keith W. Ross, *Computer Networking: A Top-Down Approach, 4th Edition*, Addison Wesley Publishing, Boston, 2008

[15] John Viega, *Practical Random Number Generation in Software*, Virginia Tech, Annual Computer Security Applications Conference, 2003.

[16] K. Pawlikowski, Joshua Jeong, Ruth Lee, "On Credibility of Simulation Studies of Telecommunication Networks," *IEEE Communications Magazine*, vol. 40, no. 1, January 2002, pp 132-139.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity (ATTN: Operations Officer)
   Camp Pendleton, California

7. Professor John Gibson
   Naval Postgraduate School
   Monterey, California